AN ARGUS SPECIALIST PUBLICATION

JULY 1986

£1.00

Your COMMODORE

YOUR BEST INDEPENDENT COMMODORE MAGAZINE



DETAILED DIRECTORIES FROM YOUR C64

WIN A VIDEO DIGITISER WITH **NEXUS**

SPLIT SCREEN ANTICS ON YOUR C16 OR PLUS/4

C128 ASSEMBLER

GET ORDERED

DISK FILING FOR YOUR C128

GET YOUR NAME ON THE LEADER BOARD

If you have ever bought another golf simulator – shoot yourself! If you are ever considering it – save yourself a bullet!
ZZAP64

Quite honestly, leaderboard makes all other golf simulations look clumsy and antiquated in comparison.
ZZAP64

This is the sports simulation of the year – if not the decade!
ZZAP64

GOLD MEDAL AWARD

TAPE £9.95 DISC £14.95

Practice Driving Range

Multiple 18 hole golf course



Amazing sound

The most realistic 3D golf simulation available

U.S. GOLD All American Software®

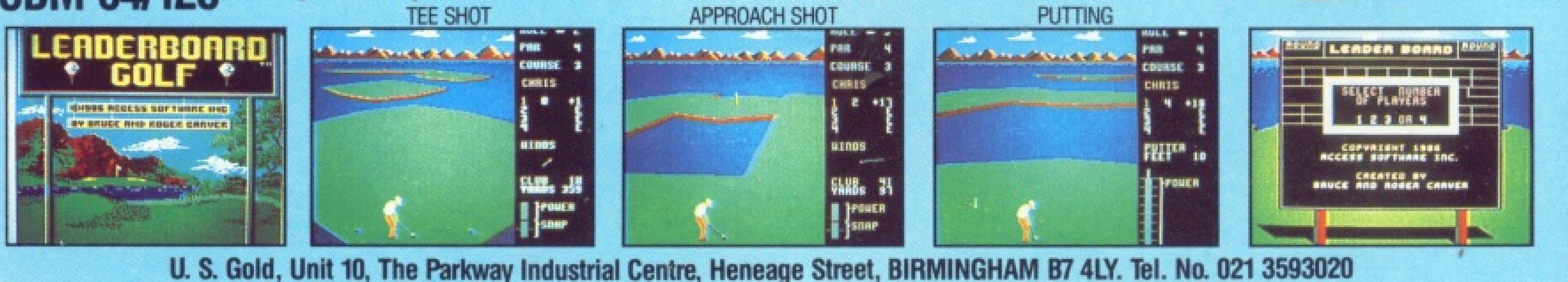
ACCESS Software Incorporated

And swing your way to a record round

Now you can become a golf pro and experience the thrill of having your name up on the "Leaderboard".
With this amazingly realistic simulation you get a true perspective view of your golf game.
As you play on different 18 hole courses on levels varying from amateur to touring professional you'll need skill concentration and control to come in under par.

Features:
- Choice of club, distance, type of shot (hook, slice, putting) and more
- Realistic Sound
- Practice Driving Range
- Joystick Controlled
- Multiple 18 Hole Golf Courses
- Automatic Scoring

CBM 64/128 Coming soon for Spectrum and Amstrad



U. S. Gold, Unit 10, The Parkway Industrial Centre, Heneage Street, BIRMINGHAM B7 4LY. Tel. No. 021 3593020

JULY 1986

VOLUME 2
NUMBER 10

Editor:
Stuart Cooke

Assistant Editor:
Marie Curry

Advertisement Manager
John McGarry

Advertisement Copy Control:
Laura Champion

Group Editor:
Dave Bradshaw

Group Managing Editor:
Wendy Palmer

Managing Director:
Peter Welham

Origination:
Ebony Typesetting

Design:
Argus Design

Editorial & Advertisement Office
No 1 Golden Square,
London W1R 3AB.
Telephone: 01-437 0626
Telex: 8811896

Your Commodore incorporating Your 64 is a monthly magazine appearing on the first Friday of each month.

Distribution by: Argus Press Sales & Distribution Ltd. 12-18 Paul Street, London EC2A 4JS. Printed by: Alabaster Passmore & Sons Ltd, Tovil, Maidstone, Kent.

Subscription rates upon application to Your Commodore Subscriptions Department, Infonet Ltd, Times House, 179 The Marlowes, Hemel Hempstead, Herts. HP1 1BB.

CONTENTS

FEATURES

SERIES

REGULARS

COMPETITION

DATA STATEMENTS

Software Shopping

Software is an area of the computer market which is nearly always booming with new products. At the moment there is a vast choice and everyone should find something to please them from utilities to the most basic of zapping games. So let's launch ourselves into the software supermarket straight away.

Something for Everyone

Those of you who have bought and enjoyed New Generation games in the past, will be interested to know that Virgin Games has now acquired all rights to New Gen's back catalogue plus the rights to seven new titles.

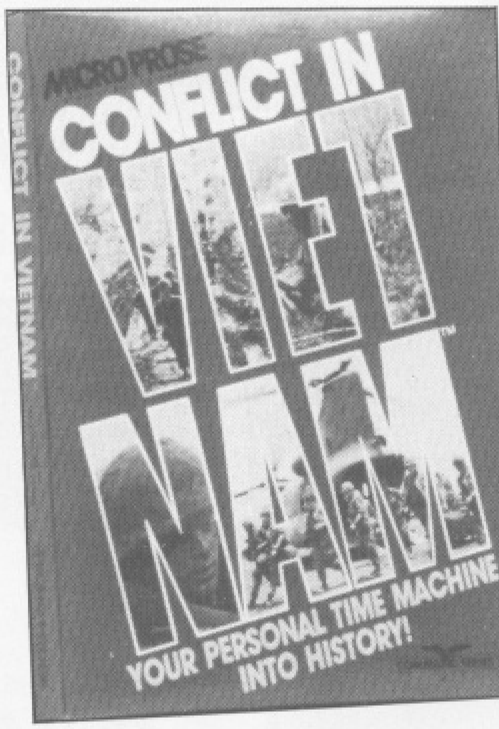
No one has yet announced which Commodore titles are going to be included in this new deal or given release dates, however Virgin's Nick Alexander was very effusive about the new deal: "I have been a fan of New Generation's software since my first days in the business when I saw their ZX81 programs at Microfairs. We've both come a long way since then and I look forward to further progress."

Chess buffs, always on the look out for a new way to indulge their regrettable addiction, should look out for Audiogenic's Grand Master Chess. This version has now been around for a while but Audiogenic recently announced that the C-16 game is now totally compatible with the Plus/4. £8.95 is the price for C-16, Plus/4 and a seperate C64 version.

Quicksilva has invented a new character – Jeremy the punk photographer. He has to wander around snapping new inventions to meet his editor's photography deadline (sounds familiar!). Your Commodore's photographic boys seem to have an easier life since they haven't yet had mutant monsters trying to nick their film! Look out for Jeremy in Quicksilva's Hocus Focus (£8.95) on the C64.

Those who like a bit of recent history intermingled with their gaming can try Microprose's Conflict in Vietnam war game. The game features events dating back to 1954 and features the decisive battles of Ia Drang (1965), Khe Sanh (1968) and Cambodia (1970). There is 110 pages of documentation to keep you occupied for hours. In the states it sells for $39.95 so you'll need to be a fairly dedicated war gamer to afford this one.

Ariolasoft is hoping you'll prefer the more leisurely pursuit of a few rounds on the green and try your hand at the new Golf Construction set. Build your own golf course or just play around on a few world famous ones which are reproduced on the program – The Belfry, Wentworth Old Course, Sunningdale and Royal St George.



An added incentive for those who become really expert is Ariolasoft's magnanimous offer of £500 worth of golf equipment for the first 12 people who get their handicaps down to scratch (zero – to us uninitiated plebs). The disk version is on sale for £14.95 and a twin cassette version is priced at £12.95.

Touch Line

Virgin Games: 95-99 Ladbroke Grove, London W11 1PG.
Audiogenic: 12 Chiltern Enterprise Centre, Station Rd, Theale, Berks RG7 4AA.
Quicksilva: Liberty House, 222 Regent St, London W1R 7DB.
Microprose: 120 Lakefront Drive, Hunt Valley, Maryland 21030, USA.
Ariolasoft: 68 Long Acre, Covent Garden, London.

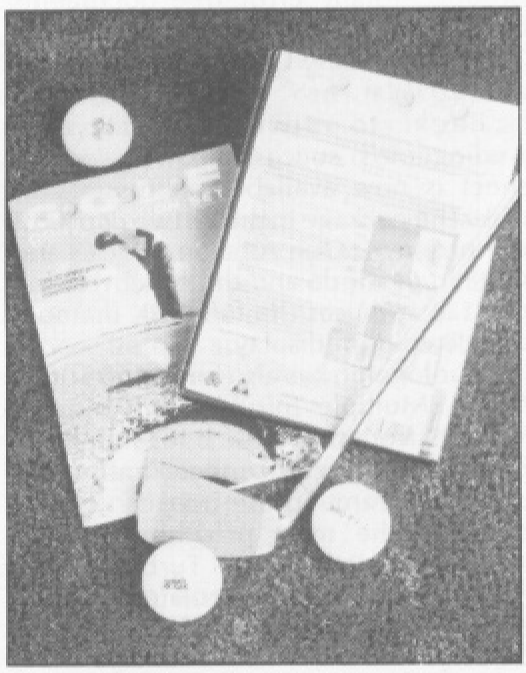
Outer Spacers

AS ALWAYS THERE ARE MASSES OF games newly available which have an interstellar setting.

Bubble Bus has come up with a cute game featuring Blob – Bio-Logically Operated Being (well, aren't we all?). The game is Starquake and Blob has to scurry through a tunnel and cave network inside a planet to collect various bits and pieces to rebuild the planet's core. There are over 500 locations and features including – Anti-grav lifts, Teleportal systems, planet surface, Security doors, sub planet exploration, propulsion pads and planetary beings (hostile). It's £8.95 on cassette for the C64.

Fans of TV's sci-fi series V will be pleased to know that Ocean has released the game of the series on C64 on cassette. For those who missed the telly program, the scenario is invaded Earth and the baddies are lizard-like aliens whose leader is called Diana! You play the part of the leader of the resistance movement, Michael Donavan. All you need to do is find out the formula for Red Dust so that you can use it to exterminate the aliens by polluting their air conditioning.

It's available now and costs a mere £8.95.



Imminently arriving in your local computer shop is CRL's new sci-fi adventure, Tau Ceti. Set on a plague-devastated earth colony in the middle of a far flung interstellar galaxy, your role is that of the intrepid suicidal maniac who has volunteered to go and repair the damage defence system of the stricken planet so that it can be recolonised. The

only way to do this is shut down the massive fusion reactor which fuels the planet.

If you want to die quickly and often it might be worth a shot at £9.95.

Touch Line

Bubble Bus: 87 High Street, Tonbridge, Kent TN9 1RX.
Ocean: 6 Central Street, Manchester M2 5NS.
CRL: CRL House, 9 Kings Yard, Carpenter's Rd, London E15 2HD.

Work, Work, Busy, Busy

Never a dull moment from Ariolasoft. This prolific software house has recently released three new utility programs.

The first – Calkit(£34.95)– is a toolkit for solving number problems. It should help you sort out the mess which is playfully termed your finances. Balance your cheque book, simplify your income tax and develop accurate home and business budgets. In other words it's a very simple to use spreadsheet.

If you use your C64 or C128 to run a business from home then perhaps Ariolasoft's second utility will be of interest to you. Entitled B/Graph, it is a visual presentation tool for sales, marketing, forecasting, accounting, management and could also have used for teachers and students. It costs £29.95 on C64/128 disk.

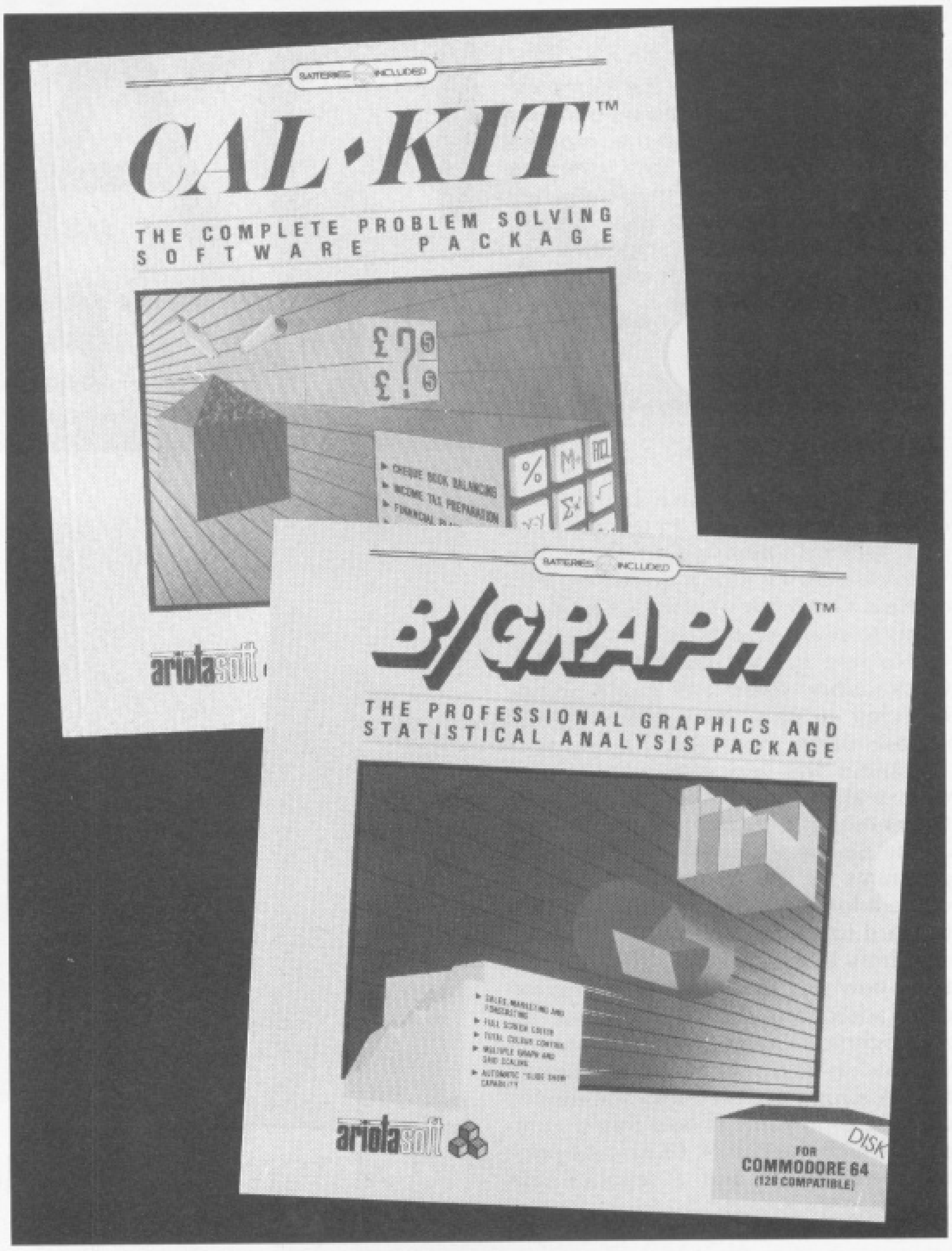
Last, but not least in the new Ariolasoft batch of utilities is Paperclip with Spellpack. It's a word processing package on C64 disk. Ariolasoft claims that it's time saving and in addition you'll produce totally error free documents because of the 15,000 word dictionary. This one's for the C64 disk and is priced at £59.95.

Back to spreadsheets and Audiogenic's successful Swift spreadsheet is now available on cassette or disk. The package includes two copies of the program. One runs on the C64 and C128 in 64 mode and the other runs on the 128 and uses the full 128K memory and 80 column display.

Audiogenic has also imported Turbo MIRV (Multiple Information Retrieval Vehicle) from the states. It loads into the 64 or 128 and runs concurrently with other programs so the user can switch between the main program and the desktop functions of Turbo MIRV. Functions include calculator, memo pad, alarm clock, calendar and auto dialler.

Touch Line

Ariolasoft: 68 Long Acre, Covent Garden, London.
Audiogenic: 12 Chiltern Enterprise Centre, Station Rd, Theale, Berks, RG7 4AA.

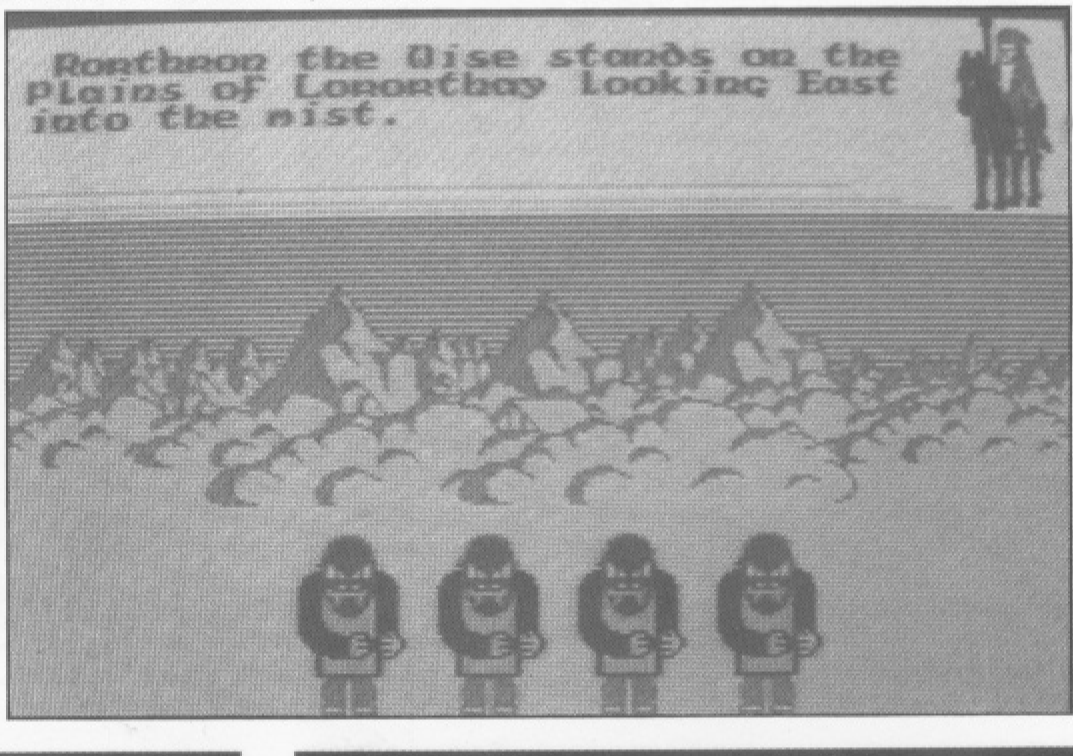


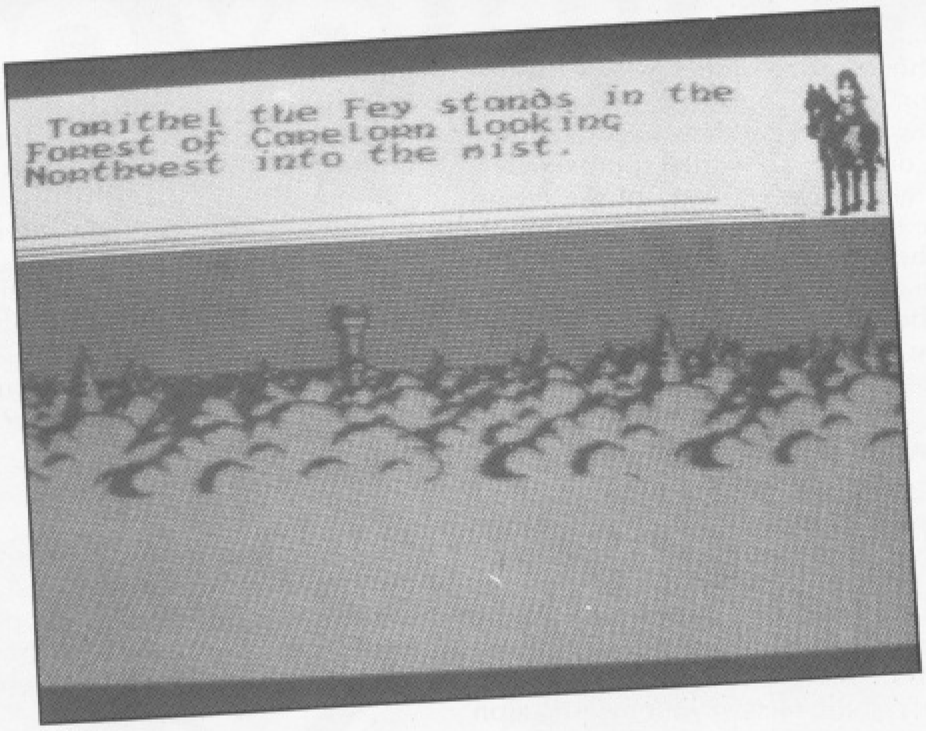
Adventure Spot

IF YOU'RE FED UP WITH DEADLY serious adventures then maybe Melbourne House's latest offering will bring a breath of fresh air to your life.

The new game is called Red Hawk and is billed as a comic strip adventure.

The hero, an ordinary guy called Kevin Oliver alternates between his normal self and Red Hawk. This miraculous transformation is brought about





when he shouts 'Kwah'. Super human skills enable him to battle against the villains and criminals in the city. It's available now and costs £8.95.

Beyond has now releasedits successful Spectrum title – Doomdark's Revenge – for the C64. It is the sequel to Lords of Midnight and contains 6,144 locations and 48,000 views!

It's an adventure cum war game set in a medieval fantasy world. It's a text adventure but there is a difference in that all possible moves can be accomplished by pressing only one key. There's a free audio cassette with the games which tells the story of Doomdark with a musical accompaniment. The price is £9.95.

Ariolasoft has also released a sequel. This one's unimaginatively entitled Archon II and – predictably – is the sequel to Archon. It's on C64 cassette and costs £9.95.

There's also a new text adventure now out from CRL. The game is called Pilgrim and it's for the C64.

It is set in the peaceful land of Meridian and you play the role of a young boy who has been given the task to go in search of The Guardian – not it's not a newspaper, it's the mysterious protector of the land. The price is £7.95.

Touch Line

Melbourne House: 60 High Street, Hampton Wick, Kingston-upon-Thames, Surrey KT1 4DB.
Beyond: Wellington House, Upper St Martins Lane, London WC2H 9DL.
Ariolasoft: 67 Long Acre, Covent Garden, London.
CRL: CRL House, 9 Kings Yard, Carpenters Road, London E15 2HD.

Alternative Taste

CRL HAS NOW RELEASED ROCKY Horror Show on C128.

The company promises that it is the most exciting version of the game to date using the extra memory available, high definition graphics, new sprites, new animation, new locations, enhanced music and new game play features. It costs £8.95.

Nu Wave, CRL's alternative software house, has announced the release of Tubular Bells for the C64. The program is claimed to be 'entertainment offering amusement for the user in both an active and passive form'. The program has two parts – a sound track and a light synthesiser. The sound track is Mike Oldfield's classic LP generated by computer.

The light synthesiser can be left to run itself or you can intervene to create your own patterns. It should be in the shops now and 795 pennies are required to buy it.

Touch Line

CRL (and Nu Wave): CRL House, 9 Kings Yard, Carpenters Road, London E15 2HD.

In Touch

COMPUNET IS GETTING INTO THE action with its own multi-user game.

The planned title is Federation II and there are over 6,000 locations. About 1000 of these will be in use at any one time. In comparison, MUD has about 400 locations.

Federation II is in the galactic trading genre and Alan Lenton, one of the authors, commented: "Just as MUD arose from dungeons and dragons, Federation II is a development of the role-playing game Traveller. It will be very different from MUD in concept."

The game is scheduled for an early 1987 launch.

Micronet has been busy lately. Interlex is now providing technical support for Micronet members.

Readers can either send their queries via electronic mail and replies can be found from page *8009007.

There is also a hotline where Micronet members can get their queries answered immediately. Micronet members can subscribe to this for £25.

Micronet has also been involved in another charity excercise.

Capital Radio's Help a London Child appeal has benefitted recently to the tune of £1300. This was raised from a frame-charged celebrity chatline.

Touch Line

Compunet: 7-11 Minerva Road, London NW10 6HJ.
Micronet 800: 8 Herbal Hill, London EC1R 5EJ.

Generally Speaking

FIRST SOFTWARE AND PUBLISHING has decided to blitz the computer industry with new launches.

There are 24 new books and nine new software packages scheduled for release this year.

There will also be general reference manuals to examine specific aspects of the C64 and C128. The Anatomy of the C128 and Tricks and Tips for the C128 are already available priced at £12.95.

Level 9 Computing is taking a stand against software piracy.

Level 9 has also begun using a Lenslok, a controversial anti-piracy device.

Lenslok tests have been placed at several places in the story of Level 9's game the Price of Magick, instead of only at the beginning and the software has been reduced to a third of its original size.

Each Lens issued has been printed with the name of the game to avoid confusion.

All Your Commodore readers who consider themselves budding professional programmers can take heart because Superior software has begun a campaign to find some new programmers.

Superior's Richard Hanson said: "This programmer recruitment drive is unique. Only a few companies have used full colour advertising for this purpose before, and, via the adverts, we are offering a free guidebook 'Top Tips for Programmers'."

Sales manager Ken Campbell added: "We're looking for programmers of all the major micros: the Spectrum, Commodore, Amstrad, Atari, BBC and Electron."

So if you want information on this campaign contact Superior of look out for the ads.

Touch Line

First Publishing: Kenilworth House, 79-80 Margaret Street, London W1N 7HB.
Level 9: PO Box 39, Weston-Super-Mare, Avon BS24 9UR.
Superior Software: Regent House, Skinners Lane, Leeds LS7 1AX.

COMPETITION

Win a digitiser from Nexus for your C64.

THIS MONTH WE'VE SET UP A competition with Nexus which may enable you to win a digitiser for your C64 plus a copy of the Nexus game.

The top prize winner, will be the first person picked out of a hat after the closing date to get the correct solution. The digitiser is worth approximately £150 and there's a copy of the Nexus game thrown in.

There will also be 24 copies of the game as runners up prizes.

The competition is based on the plot of the Nexus game to give you a taster of what you could win.

How to Enter

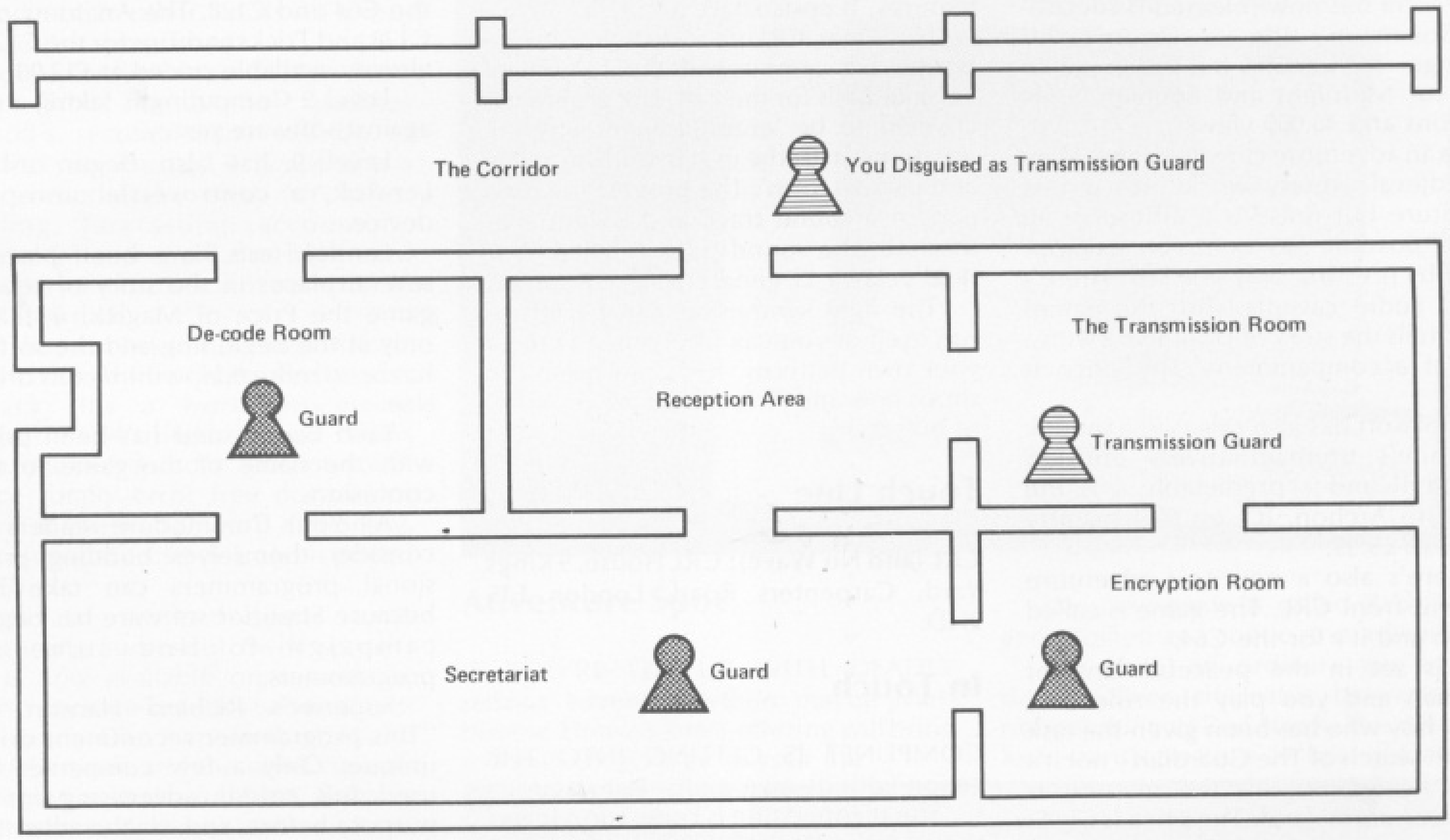
Study the diagram on this page and then carefully read the following instructions.

1. You are standing in a corridor of the drugs HQ. Your objective is the transmission room from where you can broadcast the facts of your investigation to the world. You are disguised as a transmission room guard.
2. Exchange places with the transmission room guard, so that YOU are in the transmission room and HE is in the corridor.
3. There is one key rule – Only one person may occupy a room or the corridor at any time. And only one character – a guard or yourself – may move per turn.
4. How many moves are required to achieve this and what are they?

Instructions

When you have solved the puzzle, fill in the entry form and attach a list of your answers to it on a plain piece of paper. Please write the number of moves you used on the back of your envelope.

Closing date: Friday 25 July 1986.



Nexus Competition
Entry Form
Name ....
Address ....
....
.... **Post code** ....
Number of moves used ....

Send your entry to: Nexus Competition, Your Commodore, 1 Golden Square, London W1R 3AB. Closing date: Friday 25 July 1986.
Please attach a sheet describing the moves you used. Please write clearly on the Entry form and your answer sheet.

The Rules

Entries will not be accepted from employees of Argus Specialist Publications, Nexus Productions Ltd and Alabaster Passmore and Sons. This restriction also applies to employees' families and agents of the company.

The How to Enter and Instructions sections form part of the rules. The editor's decision is final and no correspondence will be entered into.


DO YOU WANT TO BE A HERO?
Biggles
The result of unique co-operation between three ace software developers and the Biggles film production company, Biggles – The Untold Story will knock you right out of the air! It's a multi-part arcade strategy game in which each part must be completed to reach your final goal.
In the air, on the rooftops, on the ground, or in the trenches
YOU CAN BE A HERO!
Coming in May
Commodore 64
£9.95 tape, £12.95 disk
Spectrum 48K
£9.95 tape
Coming in June
Amstrad CPC
£9.95 tape, £14.95 disk
MIRRORSOFT
Purnell Book Centre, Paulton, Bristol BS18 5LQ

MISSIVES

Your letters continue to flood in, keep them coming so we know what you want.



Decline and Fall

IT CAN BE SAFELY SUGGESTED THAT the home computing industry is primarily aimed at those who play computer games and those who wish to develop programming skills and the usage of their machine. However, there appears to be a growing contradiction between these ideas and the content of available publications.

In response to Allen Webb's letter (April 86, Your Commodore), I find it increasingly difficult to find fault with his synopsis that the home computing market is in decline. Virtually all of the British home computing magazines, Your Commodore excepted, offer a diminishing amount of real interest to the average key bashing fanatic.

As an example, one magazine, although not specifically Commodore orientated, has always been able to offer a cross section of professional programs and utilities for the discerning reader. Unfortunately this format appears to have been re-directed towards more advertising, previews and reviews for hardware and software. Although I understand the need for publication to advertise, and indeed the value of in-depth reviews, I find it difficult to see why this must be achieved at the expense of actual programming (the very basis of home computing). Surely a lack of programs, hints, utilities etc. will bring into question the concept of value for money. Subsequent loss of readers will inevitably lead to loss of revenue from advertising until eventually...?

Although I have now joined the ICPUG, I have recently found the need to purchase American magazines such as Compute! and RUN to satisfy my hunger for new ideas and information. Both of these magazines offer good quality diverse programs whilst still advertising. The problems with this situation are that firstly, these magazines are wildly expensive (approx £3) and secondly, in my heart of hearts I would much rather purchase an English monthly that can compete, if not surpass, the American competition.

As many magazines have either disappeared or are on the brink of oblivion, I shall watch future editions of Your Commodore with apprehension and perhaps a little hope. At present I am generally pleased with the quality of the magazine and writers such as Allen Webb continue to perpetrate this quality. Your Commodore appears to be the last bastion of sensibility for home computing enthusiasts or are there changes on the way that I should fear?

L. Lack
Manchester

Soft Sale

REGARDING THE SOFTWARE FOR SALE offers in recent Your Commodores, might I enquire as to which programs are on the MAY YCMAR86 cassette? Are readers expected to purchase these tapes not knowing exactly which programs they will receive.

Could you indicate at the end of a program or article whether or not it will beon the Software for Sale cassette. I am sure you could do this without taking up too much valuable space.

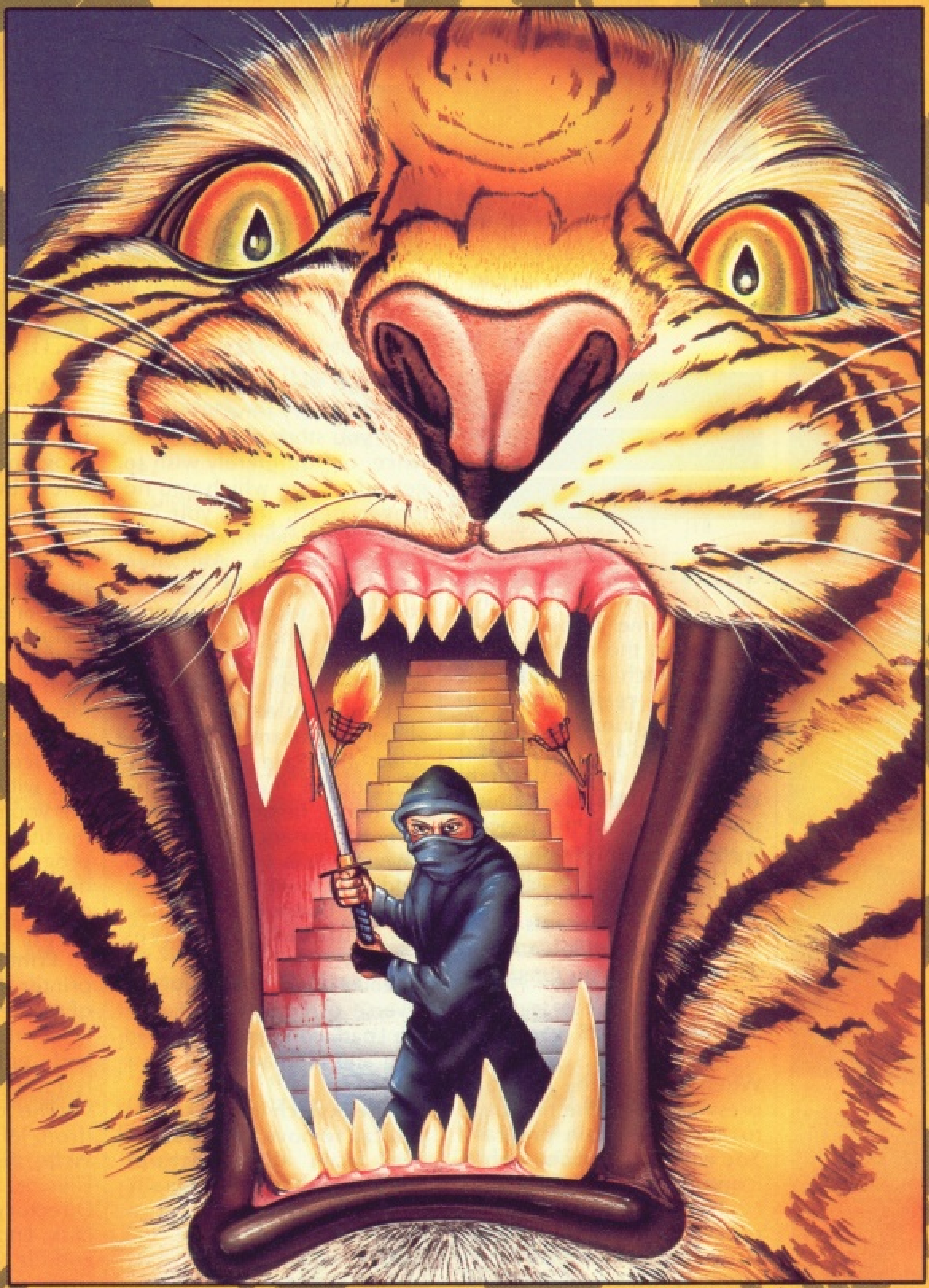
I am interested in purchasing the May cassette providing the program, Wordprok, by Ian Murray will be included.

James O. Yarker,
Pickering

Thank you for your interest, James. We have found that our cassette offer has been immensely popular, but we also appreciate your problem. Firstly, the Wordprok program is included on the May cassette. All C64 and C128 programs featured in each issue of Your Commodore are included on the cassette for the relevant month. We are looking at a system whereby we can put a small logo on certain pages to indicate whether the article is one selected for the cassette that month.

Unfortunately we cannot as yet supply C-16 and Plus/4 programs in the same way but we are looking into the possibility of this.

THE WAY OF THE TIGER



Enter the world of Avenger, a Ninja warrior of unparalleled skills and deadly powers, as he battles the forces of evil in defence of his faith and protection of the weak.

Be calm and stay silent as the outstanding animation and unrivalled combat routines take you to levels of action you'd never have thought possible.

Experience the stunning effects of triple scrolling action as you master the techniques of Hand to Hand combat, Pole Fighting and the skills of the Samuri Sword.

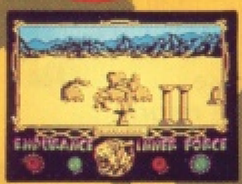
And when you believe you've succeeded in overcoming all the odds, the next in this thrilling series of adventures will beckon you forward to a further challenge of death.

Cassette £9.95

Disc £14.95

MSX 64K Spectrum 48K Amstrad CBM 64/128

Gremlin Graphics Software Limited, Alpha House, 10 Carver Street, Sheffield S1 4FS. Tel: 0742-753423





TRIPLE-SCROLL





Screenshots from Spectrum 48K

DISK BASE 128

Frank Tout helps you to clear up your litter and file it safely.

ONE PROBLEM WHICH THE vast majority of disk drive owners come across sooner or later, is the massive confusion which can build up when you realise that you have large numbers of disks lying around and you have no idean what's on any of them. When it comes to finding a specific file then the process is annoying and tedious.

Help is at hand, with this program – Disk Base 128. It will store the contents of up to 300 disks on file and help you find what you need.

What It Does

When you run the program, press 'space' at the title page to enter the main program. You will then be faced with 11 icons and a flashing cursor.

To place disk file-into the program's memory move the cursor (using the cursor keys – left and shift left) over the Write File icon and press return. You will then be asked for a field number, 1-300. Select your choice and then insert the disk which you want to store in memory in the disk drive. Then, press Return. The program will then load the directory and store it.

When this is complete, press Return to get back to the menu. The contents of that disk are now stored under the field number which you selected.

You may now repeat the process as many times as you wish, just increment the field number every time.

If you store the same disk twice, or you merely wish to get rid of one which you no longer need, then you can delete it by selecting the Erase File icon. In this case, when asked which field number you wish to erase, simply make your choice and key it in, press Return and the file is then erased.

When you've finished and all your disks are in memory, select the Save File option. You should then enter the number of fields you wish to save. For instance if you only have 40 disks in memory in fields one to 40, then you can just save that number of files. It's quicker and saves memory.

Load files is simply the reverse of Erase Files. Select the Load icon, enter the field limit and file name and press Return.

The Scratch icon is so that you can remove an unwanted file from disk without breaking out of the program. Select the Scratch icon, press Return and enter the file name to be erased. It's as easy as that.

The Directory Icon will get the directory of a disk and display it on the screen but it will not store it in memory, it's just to check and view disks without leaving the program. Select the Dir icon and press Return, the directory will then load, press space to return to the menu.

New disk is self explanatory.

View files icon will display the contents and/or titles of disks in memory. Select View icon and press Return. Then you can select TITLES or FILES by pressing T or F.

When selecting titles enter the field limit you wish to examine (1-XXX). The program will then list all the titles in memory in that section. Press Return to return.

When selecting Files enter the field number you wish to look at and press space, that will then be displayed to you. Press space to return.

If you know you've got a file somewhere but can't seem to find it select the Find File Icon and press Return. You will then be asked what file you're looking for. Enter the title and press Return. The program will then search the fields and list any which contain your lost file. Press to return.

Print files will list to the printer the title and contents of a field of your choice. Select the icon, press Return and enter the field you require to be printed.

The icons available to you (from left to right) are: Write File, Erase File, Save Files, Load Files, Scratch file, Directory, New Disk, View Disk, End Program, Print File. End file takes you back to the title page, the disks in memory will not be lost.

The Program

The program is REMmed so you can follow it and the data statements are for ICONS.

Variables

FE – field
di$(x,x) – Dim/array for disk contents
c$(x) – Dim array for titles
a$(x) – arrays for icon data
P$ – lower case
PP$ – carriage return

PROGRAM: DISKBASE

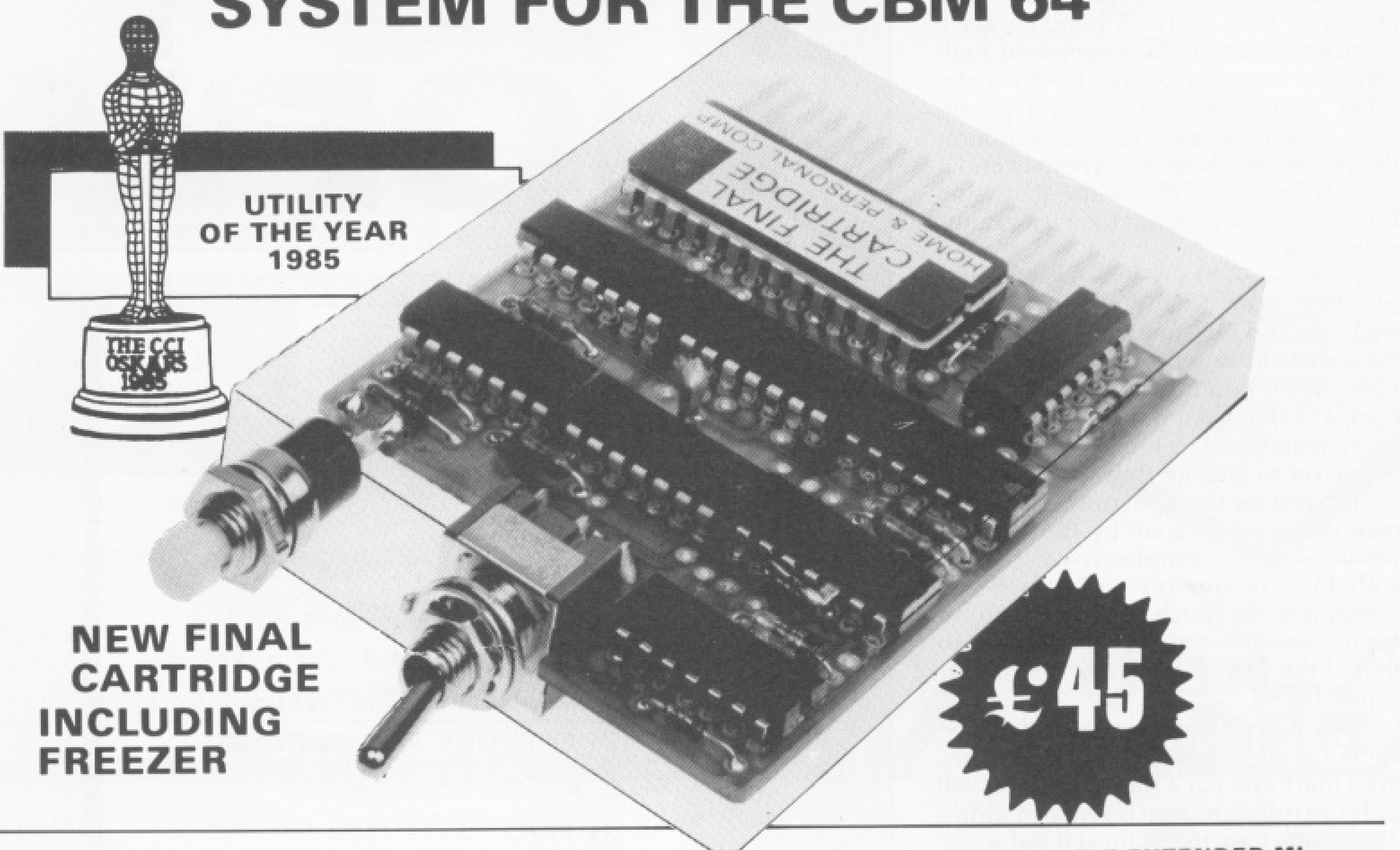
```
0 poke53280,0:poke53281,0:cl
r:fast:dimdi$(300,60):dimc$(
60):p$=chr$(14):pp$=chr$(13)
:printp$
2 gosub286
4 goto258
6 color0,12:color1,1:color4,
13:graphic2,1,12:window0,12,
39,24:poke 53281,15:printchr
$(144):slow
8 v=8:fort=1to8:gshapea$(t),
v,4:v=v+28:next
10 fort=1to3:gshapeb$(t),v,4
:v=v+28:next
12 char1,0,5,p$+"
    Command.
",1
14 char1,10,7,p$+"
        ",1:lo=11:v=312
16 ifda=1thenchar1,0,11,"***
*data in memory****data in m
emory****",1
18 sprite4,1,1+rnd(1)*15,1:m
ovspr4,v,54
20 getz$:ifz$=""then18
22 ifz$="[LEFT]"andv>36thenv
=v-28:lo=lo-1
24 ifz$="[RIGHT]"andv<312the
nv=v+28:lo=lo+1
26 ifz$=pp$then30
28 goto18
30 onlo goto34,84,206,240,90
,110,124,156,186,258,224
32 end
34 gosub452:input"field No:-
";fe
36 char1,5,7,p$+" Write File
 to Base ",1:print"[RVSON]Fe
ild ";fe:sleep3
38 printchr$(144)chr$(147)
40 open2,8,15
42 open1,8,0,"$0"
44 get#1,a$,b$
46 get#1,a$,b$
48 get#1,a$,b$
50 c=0
52 ifa$<>""thenc=asc(a$)
54 ifb$<>""thenc=c+asc(b$)*2
56
56 print"[RVSON]"mid$(str$(c
),2);tab(3)"[RVSOFF]";
58 get#1,b$:ifst<>0then74
60 ifb$<>chr$(34)then58
62 get#1,b$:ifb$<>chr$(34)th
enprintb$;:di$(fe,x)=di$(fe,
x)+b$:goto62
64 get#1,b$:ifb$=chr$(32)the
n64
66 printtab(18);:c$(w)=""
68 c$(w)=c$(w)+b$:get#1,b$:i
fb$<>""then68
70 print"[RVSON]"left$(c$(w)
,3):x=x+1:w=w+1
72 ifst=0then46
74 close2:close1:printchr$(1
4)chr$(149)chr$(147):printdi
$(fe,0):printchr$(31):fort=1
tox:printdi$(fe,t);:printtab
(20)c$(t):next:w=0:c$(w)="":
da=1
76 poke4097+fe,x:x=0
78 print"          Press Spac
e When Ready"
80 getq$:ifq$<>" "then80
82 printchr$(147):graphic2,0
,12:window0,12,39,24:goto12
84 char1,10,7,p$+"   Erase F
ile.     ",1:sleep3
86 input"Field No:- ";fe
88 fort=0to50:di$(fe,t)="":n
ext:print"  File In Field "
;fe;" Now Erased":sleep4:pri
ntchr$(147):goto14
90 rem**scratch file
92 char1,10,7,p$+"   Scratch
 File     ",1:sleep3
94 gosub452
96 input"File Name";na$
98 open15,8,15:print#15,"s0:
"+na$
100 print"        File ";na$;"
 Now Erased"
102 close15
104 gosub450
106 geta$:ifa$<>" "then120
108 gosub454:goto14
110 rem**dir**
112 char1,10,7,p$+"       Dir
ectory    ",1:sleep3
114 gosub452
116 directory
118 gosub450
120 geta$:ifa$<>" "then120
122 gosub454:goto14
124 rem**new disc**
126 char1,10,7,p$+"     New
Disc     ",1:sleep3
128 gosub452
130 print" This Erases any D
ata Stored on Disc ??"
132 input"ARE YOU SURE";qu$
134 ifqu$="n"then154
136 ifqu$="y"then140
138 goto142
140 goto132
142 input"File Name";na$
144 input"Identity(two numer
ics)";id
146 headerna$,iid,d0,u8
148 print"     Disc Now Read
y And Formatted"
150 gosub450
152 geta$:ifa$<>" "then152
154 gosub454:goto14
156 rem**examine files
158 char1,10,7,p$+"  Examine
 Files.    ",1:sleep3
160 printchr$(147):input"[BL
ACK]Disc [RVSON]T[RVSOFF]itl
e's or [RVSON]F[RVSOFF]iles"
;w$
162 ifw$="t"then168
164 ifw$="f"then174
166 goto156
168 gosub452
170 input"Field Limit";fe
172 fort=0tofe:printdi$(t,0)
;:printtab(30);t:next:goto18
0
174 gosub452
176 input"Field No:-";fe
178 printdi$(fe,0):fortt=0to
peek(4097+fe):printdi$(fe,tt
):nexttt
180 gosub450
182 geta$:ifa$<>" "then182
184 printchr$(147):graphic2,
0,12:window0,12,39,24:goto14
186 rem**sort
188 char1,10,7,p$+"     Sor
t File     ",1:sleep3
190 printchr$(31):input"What
 file Are You Looking For,";
kn$
192 print" Please Wait While
 I Look"
194 :fort=0to300:fortt=0tope
ek(4097+fe)
196 ifkn$=di$(t,tt)thenprint
"File Found in Field ";t
198 nexttt,t
200 gosub450
202 geta$:ifa$<>" "then202
204 gosub454:goto14
206 rem**save to disc
208 char1,10,7,p$+"     Save
 Files     ",1:sleep3
210 gosub452
212 input"Field Limit:-";fe
214 input"File Name";na$
216 open8,8,8,"0:"+na$+",p,w
":fort=0tofe:ww=peek(4097+t)
:print#8,ww:fortt=0topeek(40
97+t):print#8,di$(t,tt),chr$
(44):nexttt,t:close8,8,8
218 gosub450
220 geta$:ifa$<>" "then220
222 gosub454:goto14
224 rem**print files
226 char1,10,7,p$+"     Prin
t Files    ",1:sleep3
228 gosub456:input"Field No:
-";fe
230 open9,4,7:cmd9:printchr$
(14)
232 printdi$(fe,0):print:for
t=1topeek(4097+fe):printdi$(
fe,t):nextt
234 print#9:close9:gosub450
236 geta$:ifa$<>" "then236
238 gosub454:goto14
240 rem**load from disc
242 char1,10,7,p$+"     Load
 Files    ",1:sleep3
244 gosub452
246 input"Field Limit:-";fe
248 input"File Name";na$
250 open8,8,8,"0:"+na$+",p,r
":fort=0tofe:input#8,ww:poke
4097+t,ww:fortt=0topeek(4097
+t):input#8,di$(t,tt):nexttt
,t:close8,8,8
252 gosub450
254 geta$:ifa$<>" "then220
256 gosub454:goto14
258 rem**title page
260 sprite4,0:color0,1:color
1,12:color2,13:color3,16:col
or4,1:graphic3,1:scale1,640,
200:width1:sleep1:slow
262 box2,140,90,220,170,45
264 box1,140,60,220,140,45,1
266 box2,140,60,220,140,45
268 draw2,96,115to96,145:dra
w2,266,86to266,116:paint2,10
8,144:paint2,98,116:paint2,2
65,87
270 fort=1to39step3:draw3,20
0-t,70+t/2to244-t,92+t/2:nex
t
272 width2:fort=0to5:draw0,1
00,125+tto150,150+t:next
274 box1,50,50,98,98,,1
276 width1:circle3,74,74,10,
10:paint3,77,77:a=1
278 char1+rnd(1)*3,0,3,chr$(
14)+"Discbase (128)By F TOUT
 1986 Press Space",1
280 geta$:ifa$<>" "then278
282 fast:goto6
284 end
286 rem
288 fort=3584to4096:reada:po
ket,a:c=c+a:next
290 fort=1to8:sprsavt,a$(t):
next
292 fort=3584to3841:reada:po
ket,a:c=c+a:next
294 fort=1to4:sprsavt,b$(t):
next
```

DISKBASE 128

```
296 ifc<>105598thenprint"data error:":end
298 fort=0to300:poke4097+t,0:next
300 slow:return
302 data255,255,255,128,0,79,128,0,159,128
304 data1,61,128,2,121,128,4,241,128,9
306 data225,128,19,193,128,39,129,128,79,1
308 data128,92,1,144,185,1,145,113,1,145
310 data193,1,147,1,1,150,65,1,152,65
312 data1,152,227,1,143,190,1,128,0,1
314 data255,255,255,0,255,255,255,128,0,1
316 data128,0,1,128,0,1,135,248,1,135
318 data252,1,135,126,1,134,63,1,132,31
320 data129,132,63,193,133,247,225,131,235,241
322 data129,149,249,128,202,253,128,101,127,128
324 data50,191,128,25,95,128,12,175,128,6
326 data87,128,3,43,255,255,255,0,255,255
328 data255,128,0,1,159,255,249,154,191,249
330 data149,95,249,154,191,225,159,231,225,159
332 data195,249,159,195,249,159,231,249,159,255
334 data249,159,231,249,159,195,249,159,195,249
336 data159,195,249,128,0,1,128,24,1,128
338 data60,1,128,126,1,128,24,1,255,255
340 data255,0,255,255,255,128,24,1,128,126
342 data1,128,60,1,128,24,1,128,0,1
344 data159,255,249,154,191,249,149,103,225,154
346 data195,225,159,195,249,159,231,249,159,255
348 data249,159,231,249,159,195,249,159,195,249
350 data159,195,249,159,195,249,159,255,249,128
352 data0,1,255,255,255,0,255,255,255,128
354 data0,1,128,0,1,157,127,57,148,92
356 data249,154,147,249,149,99,225,159,132,97
358 data158,67,137,153,195,241,135,231,137,159
360 data252,121,159,227,249,159,3,249,152,195
362 data241,135,195,137,159,192,121,131,31,249
364 data128,0,1,128,0,1,255,255,255,0
366 data255,255,255,128,0,1,158,125,249,147
368 data17,25,145,17,249,147,17,97,158,125
370 data57,128,0,1,159,231,225,159,195,249
372 data159,195,249,159,231,249,159,255,249,159
374 data231,249,159,195,249,159,195,249,159,195
376 data249,159,195,249,159,255,249,128,0,1
378 data255,255,255,0,255,255,255,128,0,1
380 data155,123,25,157,67,25,149,114,73,151
382 data67,89,155,121,241,128,0,1,159,231
384 data225,159,195,249,159,195,249,159,231,249
386 data159,255,249,159,231,249,159,195,249,159
388 data195,249,159,195,249,159,195,249,159,255
390 data249,128,0,1,255,255,255,0,255,255
392 data255,128,0,1,128,0,1,128,0,1
394 data128,126,1,131,129,193,156,0,57,160
396 data126,5,131,195,193,159,60,249,190,102
398 data125,254,90,127,190,102,125,159,60,249
400 data131,195,193,160,126,5,156,0,57,131
402 data129,193,128,126,1,128,0,1,255,255
404 data255,0,7
406 data255,255,255,128,0,1,128,0,61,191
408 data255,253,191,254,29,160,1,229,191,255
410 data253,191,240,253,160,15,5,191,255,253
412 data191,135,253,160,120,5,191,255,253,188
414 data63,253,163,192,5,191,255,253,161,255
416 data253,158,0,5,191,255,253,128,0,1
418 data255,255,255,0,255,255,255,128,0,1
420 data159,255,249,152,0,1,159,240,1,152
422 data0,1,159,255,249,128,0,1,159,128
424 data25,153,240,25,152,60,25,152,15,153
426 data152,1,249,128,0,1,159,255,225,152
428 data0,57,152,0,25,152,0,57,159,255
430 data225,128,0,1,255,255,255,0
432 data 255,255,255,128,0,81,128,0,65,128,15,209,128,15,193,128,11,209 , 128 , 9 , 193 , 129
434 data250,209,129,248,127,129,122,21,129,56,5,191,90,21,191,15,253,175,66,129,167,0,129,171,66,129,161,255,129,168,80,1,160,16,1,168,80,1,255,255,255,0
436 data0,0,0,31,255,252,32,0
438 data2,39,255,242,40,0,10,41,255,202
440 data42,0,42,42,127,42,42,128,170,42
442 data156,170,42,162,170,42,162,170,42,156
444 data170,42,128,170,42,127,42,42,0,42
446 data41,255,202,40,0,10,39,255,242,32
448 data0,2,31,255,252,0,0,255
450 print"       Press Space To Continue.":return
452 sprite4,0:printchr$(147):graphic0,1:window0,0,39,24:print" Press No Scroll To Stop and Continue.":window0,1,39,24:return
454 printchr$(147):graphic2,0,12:window0,12,39,24:return
456 sprite4,0:printchr$(147):graphic0,1:window0,0,39,24:print" Please Ensure Printer Is Switched On.":window0,1,39,24:return
```

THE FINAL CARTRIDGE ®

THE FIRST OUTSIDE OPERATING SYSTEM FOR THE CBM 64 *



NEW FINAL CARTRIDGE INCLUDING FREEZER

This new operating system built in a cartridge does not use any memory and is always there. Compatible with 98% of all programs.

DISK TURBO - 6 times faster loading - 8 times faster saving.

TAPE TURBO - 10 times faster, even with files - normal Commodore commands - compatible with standard turbo's.

ADVANCED CENTRONICS INTERFACE - compatible with all the well-known centronics printers and Commodore printer programs. Prints all the Commodore graphics and control codes (important for listings).

SCREEN DUMP FACILITIES - of low-res Hi-res and multicolour screens!! Prints full page with 12 shades of grey for multicolour pictures even from games and programs like Doodle, Koala pad, Printshop etc. Searches automaticly for the memory-address of the Picture. Special version available for the CBM 801 and 803 printers.

24K EXTRA RAM FOR BASIC-PROGRAMS AVAILABLE: Two new commands ,,Memory read'', ,,Memory write''. They move 192 bytes with machinelanguage-speed anywhere in the 64K Ram of the CBM 64. Can be used with strings and variables.

BASIC 4.0 COMMANDS - like Dload, Dsave, Dappend, Catalog, etc.

BASIC TOOLKIT - with Auto, Renum (incl. Goto and Gosub), Find, Help, Old, etc.

* works with C128 in the 64 mode.



Original multicolour full page screen dump print out.

PREPROGRAMMED FUNCTION KEYS: - Run, Load, Save, Catalog, Disk commands, List (removes all list-protections).

KEYBOARD EXTRA'S - Allows you to delete part of a line; stop and continues listings; move cursor to lower lefthand corner. Pokes and Syscalls in Hex. Typ-command operates your printer as a typewriter.

COMFORTABLE EXTENDED ML. MONITOR: - with relocated load scrolling up and down. Bankswitching, etc. - **does not reside in memory.**

RESET SWITCH: - resets to monitor; resets with old, resets to Hi-res printing; resets every protected program.

ON/OFF SWITCH - we hope you never need that one.

FREEZER:

Stops and continues almost every program and allows you to make a total back up to disk or tape automatically.

Specs: **Creates one file on disk or tape.**
Packs the program.
Freezes 4 to 6 times faster than dedicated freezers.

Menu driven:
Freezer options include:
full page printing
fore- and background colour changes
reverse printing
jumps to monitor or reset

Training mode:
kills sprite collision detection.

12 Months replacement guarantee.

Home Personal COMPUTERS

U.K. ORDERS Available by the wellknown Commodore Dealers or directly from
H & P Computers
9 Hornbeamwalk
Witham Essex CM8 2 SZ England d.
Telephone: 0376 - 51 14 71.

Stuart Cooke takes a look at a sub £50 printer.

Cheap COPY

NO MATTER WHAT YOU ARE DOING with your computer there will be a time when you will need to get some sort of printout. If you are writing your own programs then you will need to get a printout so that you can check it through carefully as you develop it. If on the other hand you are copying a listing, such as one from Your Commodore, then errors are bound to creep in and it is much easier to check the magazine listing against a printout than it is against your TV set.

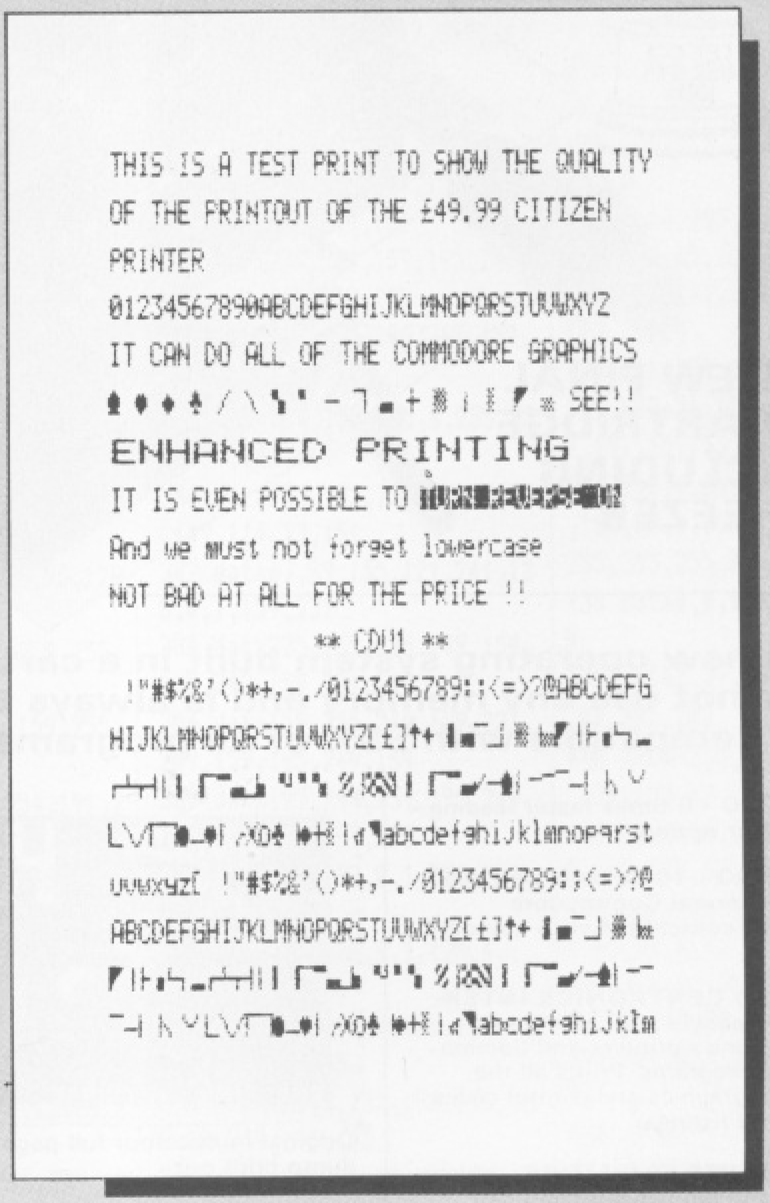
However, there is one small problem, cost. Nearly all printers are well over the £100 mark, in fact it is very difficult to find a printer that is under £200. Well don't worry, Spectrum, a company that has a retailer in just about every high street, is importing a printer that is set to change this.

The printer that Spectrum is selling is manufactured in Japan by Citizen. It is designed to be completely compatible with the Commodore range of home computers. So much so that they have even made the case a similar colour to that of the C64. The price of this printer is just £49.99.

Obviously there are bound to be some corners cut with a price like this. Well there are. For a start the printer will only use roll paper that is 80mm in width. This is very easy to obtain as it is the size of paper used by many desk top calculators. Secondly the printer will only print a maximum of 40 characters per line. An example printout is included (full size) with this article so that you can judge for yourself the actual quality of the print. My feeling is that the manufacturers claim that the printer 'Expands your Commodore into a word and data processing system' is a little wide of the mark. I couldn't see myself sending business letters out on paper that is only 2¾ inches wide. Even so it does work well and if you don't already own a printer then it is well worth looking at. In fact a second printer of this quality is quite often useful to have around for producing quick directory listings etc.

Even though the printer does have a few 'faults' it has a surprising number of points in its favour. The printer measures only 240mm by 176mm. This means that it takes up very little room, in fact the review model is sat on top of my monitor.

Even though it is small and cheap it also has a large number of commands available. All Commodore graphic characters can be printed so there are no problems reading listings. Some printers totally ignore the Commodore graphic codes. It is possible to print out listings in lower case mode as well as in graphics mode. It is also possible to produce user defined graphics on the printer.


THIS IS A TEST PRINT TO SHOW THE QUALITY
OF THE PRINTOUT OF THE £49.99 CITIZEN
PRINTER
01234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ
IT CAN DO ALL OF THE COMMODORE GRAPHICS
♠ ♥ ♦ ♣ / \ ... SEE!!
ENHANCED PRINTING
IT IS EVEN POSSIBLE TO TURN REVERSE ON
And we must not forget lowercase
NOT BAD AT ALL FOR THE PRICE !!
** CDU1 **
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFG
HIJKLMNOPQRSTUVWXYZ[£]↑←
...abcdefghijklmnopqrst
uvwxyz[!"#$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[£]↑←
...abcdefghijklm

Enhanced printing is also catered for and one unusual feature is the fact that the ribbon is both black and red. Unfortunately you can only use one colour per line. Even so this is extremely useful for highlighting text.

Even though quite a bit of compatability is there, there are many programs that will not work correctly with this printer. You should be OK with a program that simply produces listings but if any graphics are printed you may get into problems. This is because most programs that print graphics or perform screen dumps assume a printer width of 80 characters.

Even so it is very difficult to complain at a printer that offers quite a large amount of compatibility with your Commodore computer at a very low price.

If you could with a printer but until now thought that the price was putting you off, take a look at this one.

Touchline
Citizen two-colour printer
Retailer — Spectrum
Price £49.99 RRP

Game of the Month

Jayne Goin spends some time on the fairway and finds it great fun.

TIGRESS MARKETING ARE VERY PROUD of the latest addition to their ventures. Released through Ariolasoft, the Golf Construction Set should prove to be the ultimate in computer golf simulation until laser disk games appear.

Whether you're a golfing pro or a rank amateur, this package has everything to recommend it. A full bag of clubs selected by yourself, a range of skills and techniques plus several world famous courses to play on.

Wentworth, Sunningdale, the Belfry and St Georges are all supplied initially and Ariolasoft promise more courses to follow in the future. The construction kit is so detailed that it is possible to recreate the special features of each course from the claustrophobic wooded appearance of Sunningdale to the numerous water hazards of the wet-look Belfry.

Golfing competitions take two forms: Match Play and Competition.

Competition is the familiar game where the player who holes out on the eighteenth green in the fewest number of strokes is the winner. In this simulation up to four players can compete.

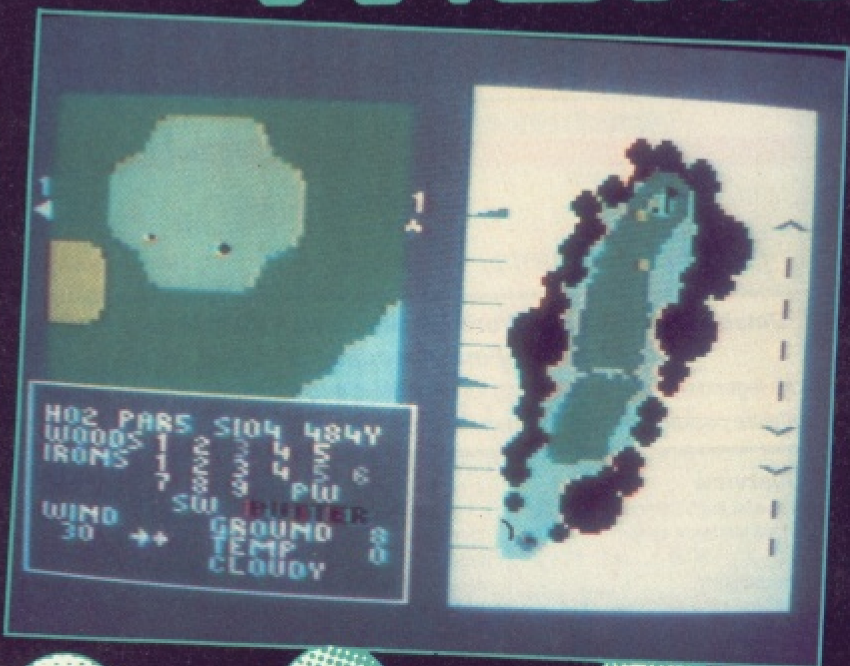
Match Play is a two player game and each hole is either won, lost of halved. This means that the winner is the first person to win 10 holes and the remaining holes are left unplayed. Normally, a drawn match is played on from the first tee until a two hole lead is established by one of the players but this is not possible in this simulation so a drawn match can occur.

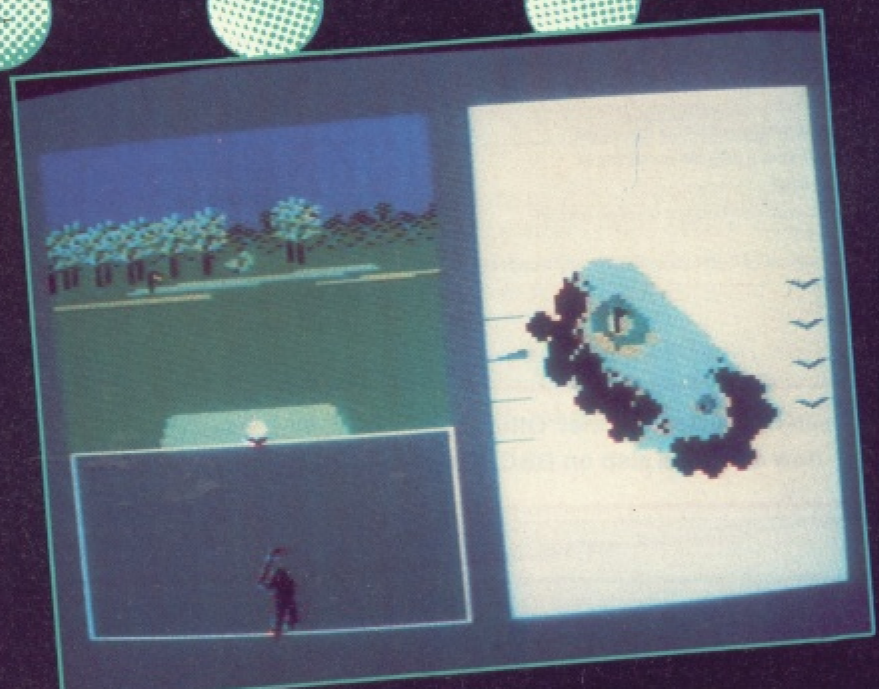
As each new hole is played it is loaded from tape to disk. Fortunately this can be described as the normal delay caused by the players walking from green to tee. As the game starts this is hinted at by the legend 'Walking to the first tee' emblazoned across the screen as the rest of the game loads after the initial option screens.

A full complement of clubs cannot be carried so three must be selected for omission at the start of the game. Playing a shot involves several further decisions. The screen design is there to assist your judgement. The top left quadrant of the screen shows the view towards the tee in 3D. Under this is a panel which gives all the vital statistics of the hole and the weather and turf conditions. The right half of the screen shows the plan view of the hole.

Weather and turf conditions affect the length of shot which can be played. Wind direction dictates a particular extra force on the ball, warm air slows







the ball less than cold air, rain has a similar effect and the dryness of the turf affects bounce and distance.

Taking all of these factors into consideration your club is chosen and the direction of the shot is determined by moving a cross in the direction of play. At this point the extremely clever nature of the 3D display becomes obvious. Each time you reposition the cross the view is redrawn. In this way a very realistic view is created.

Next the amount of loft and fade is selected. Loft determines whether the ball will fly high or skim across the grass. Fade causes the ball's trajectory to bend in a deliberate way unlike slicing or hooking which are caused by incorrect handling of the club.

The final decision is the actual stroke. A small golfer appears at the bottom of the now cleared text screen and proceeds to take swings at the ball. Pressing the fire button ono the backswing causes the golfer to hit the ball from his current position. The higher his backswing the harder the ball is hit.

If the ball lands in an unplayable position you have the option at the start of each stroke selection sequence to move back to your ball's previous position, end the game or proceed immediately to the next hole.

On reaching a distance within a few yards of the flag, the 3D screen gives way to a large scale plan view of the green showing the vertical and horizontal components of the slope of the ground. Allowing for the roll which this slope will cause, you place your targeting cross and swing. If you manage to hole out before reaching this plan view you are treated to a slow-motion, action replay of your shot.

As in the real game each player has a handicap. Starting as a novice your handicap is 28 and you must prove your worth as your play each round. The handicap is designed to smooth out the differences between professionals and amateurs. At the end of a Competition game, your handicap is subtracted from the total number of strokes taken and this gives your final score. If this value minus the total par for the course is less than your handicap value, a new grading is generated and this lower handicap can be saved to tape or disk.

In Match Play each hole is allotted difficulty level in comparison with the other holes. This means that each course has a fully nominated scale of one to 18, each hole having a unique value. This is taken into account in conjunction with your handicap when deciding which player has won.

The construction section is very easy to use if the correct procedure is followed. After choosing the par value for the hole, the tee and flag are placed using a joystick and a screen readout of the distance between them.

Next the boundary line is drawn and then the elements of the course can be added. The choice is made via two icon menus, one for the size of object and the other for the type. The types available are water, bunker, green, fairway, rough, woodland or scrub. The size gauges are roughly circular but range from about four characters in area to almost pixel size so most shapes can be created.

After entering the relative slope on the course and the stroke index, the hole is complete.

I loved this game and I know that the Chairman and the Pro at Wentworth feel the same way about it. If this is still not recommendation enough then just try the game, few will be able to resist its immense appeal.

Evesham Micros

THE UTILITY SPECIALISTS

Quickdisc+
FAST LOAD PLUS UTILITY CARTRIDGE

After enjoying considerable success since its release we have now made some improvements to "QUICKDISC+" to maintain its position as the best value in the disc "speed up/utility" cartridges.

Fast LOAD (now 5-7 times normal speed). Works with majority of protected software. Can be switched in and out from the keyboard.

Fast SAVE at 7 times normal speed.

Fast Format takes just 20 seconds.

Fast Backup copies an entire disc in four minutes (not protected software).

Very Fast File Copier for selective file copying at HIGH speed. Now handles files up to **248 blocks** long.

Improved DOS commands (DOS 5.1) makes for easy use of the disc drive. eg. $(RETURN) will LOAD and display a directory without overwriting BASIC. SHIFT RUN/STOP will LOAD "0:*",8,1 etc. **Very, very useful.**

Incorporates Centronics printer software (user port) **with CBM graphics** capability (requires user port centronics cable).

A RESET switch is fitted. (We have found this to be "unstoppable", it even preserves the tape buffer).

NO MEMORY IS USED by this cartridge, it is totally "transparent" and uses special switching techniques.

"128" and 1570 compatible in '64 mode.

PLUS MANY MORE USEFUL FEATURES TO MAKE YOUR 1541 FAR MORE FRIENDLY TO OPERATE.

ONLY £19.95

When reviewing "Quickdisc+" Commodore Horizons said: **"A MUST FOR EVERY 1541 OWNER".**

NOW EVEN FASTER

DISC DISECTOR V4.0

Disc-Disector

Now the **fourth generation** of the country's **leading** disc back-up/utility package is available. Various improvements have been incorporated to re-establish "Disc Disector" as the **ultimate weapon.** Of particular importance are the following.

"EVESHAM 3 MINUTE NIBBLER" is the **latest version** of the infamous "Evesham Nibbler" now boasting **even more power and speed. Will now copy even the most highly protected discs in 3-4 minutes.** Handles the latest types of disc protection **completely automatically.** At the time of going to press this program copied **virtually every disc program available for testing including the latest in games and business software.** (It even copies '128 format or discs in CPM mode on the '128).

"DUAL DRIVE NIBBLER" allows the use of two drives to make **even faster back-ups.** Incorporates device number change software so no hardware mods necessary.

"NOVA TRANSFER" will **transfer to disc** the latest Nova Load programs including **multi part loaders.** A very useful utility that also includes routines to transfer "Summer Games II" and "Beach-Head II" to disc.

"DISCO" will transfer standard speed load software to disc.

"TRANS QD" allows you to save a fast load system to your own discs.

"DOUBLE BACK UP" is a **very fast** two drive back up.

As well as these important newcomers all the old favourite utilities which have helped earn "Disector" such a large following are included. These include **Menu Maker (selective), Fast Format, Unscratch, Disc Monitor, Disc Orderly, Fast File Copy, Index, etc. etc.**

The whole package is menu driven and has its own fast boot system built in for your convenience. Pays for itself the first day your receive it.

ONLY £29.95

Customers with earlier versions may return them along with a payment of £9.95 to receive V4.0

All programs are CBM 128 and 1570/71 compatible in '64 mode.

QUIET DRIVE STOPS

This package incorporates new drive stops for **two 1541s** that will **end for good** the dreaded "woodpecker". The fitting process is very easy and a test program is provided to check the fitment. Helps prevent future alignment problems and makes your drive purr with satisfaction.

ONLY £4.95

DOUBLER

The tape back up "device"

Doubler has been an **enormous success** and continues to sell very well. Why? Because it is the **best product of its type** on the market. Doubler uses a unique method that copies **all types of software** regardless of speed. It consists of **hardware and software**. The software is the key part as unlike other products of its type Doubler creates a **brand new machine copy**. It is very easy to use and very successful. In fact our tests **have proved** that this "device" can achieve **100% success**.

Requires access to two data recorders.

ONLY £12.95

Why settle for less—this is the best.

FREEZE FRAME

THE NEW GENERATION OF BACKUP METHODS IS NOW EVEN BETTER

THE LAST WORD IN BACKUP TECHNOLOGY

"Freeze Frame" has enjoyed **incredible success** for the past few months. Sales both in the U.K. and abroad have been very good and **thousands of satisfied customers** are using them.

Continual development has enabled us to launch an **improved version** of the product that incorporates a few useful improvements **PLUS** it is **even more powerful.** At the time of going to press our **thorough testing could not find one single, memory resident, program that "Freeze Frame" could not handle.** (This includes **ALL** the latest software up to 4th May 1986).

OPERATING "FREEZE FRAME"

"Freeze frame" is simplicity itself to use. It is cartridge based so is simply plugged into the cartridge port. When the computer is switched on a message is displayed, pressing "RETURN" will clear the computer back to the normal start up screen. Software can now be loaded from tape or disc **completely as normal.** The latest version of "Freeze Frame" will, to the best of our knowledge, allow **ANY software** to load and run normally (unlike competitive products).

"Freeze Frame" can be brought into operation at any convenient point by pressing the button on it. You can then do one of three things:—

1. **Pressing "D" will save a working version of the program in memory to a formatted disc. This version will include a high speed reload.**
2. **Pressing "S" will do the same as "D" except that the reload will be at standard 1541 load speed. (Also suitable for U.S. spec. machines, 4040 drives, some fast load systems etc.)**
3. **Pressing "T" will save a working version of the program in memory to tape. This incorporates a high speed reload at approx. 2400 baud.**

THAT IS IT!!

ALL PROGRAMS SAVED WITH "FREEZE FRAME" CAN BE RELOADED INDEPENDANTLY OF THE UTILITY

WE ARE CONFIDENT THAT FREEZE FRAME IS THE MOST POWERFUL AND FLEXIBLE BACKUP SYSTEM IN THE WORLD.

Does any other utility offer all this?

TAPE TO DISC — **DISC TO TAPE**
DISC TO DISC — **DISC TO TAPE**

PLUS the added flexibility of being able to save at any stage and return to that position, because programs restart from the point where the button is pressed.

An awe inspiring success rate.

Customise programs by putting in POKEs for endless lives etc. and then save with those active.

Reload independant of the cartridge.

Available now.

"FREEZE FRAME" HAS NO SERIOUS COMPETITION.

COMPETITIVE PRODUCTS

A successful product always spawns imitators and "Freeze Frame" is certainly no exception. Fortunately for us, but perhaps not for the people who have purchased them, their main similarity seems to be in advertising style.

"The Final Cartridge" is a good product in many ways, **BUT** when it comes to making backups it is just not in the same class as "Freeze Frame". Another major drawback, not made clear by advertising is that saved programs can **ONLY be reloaded with the cartridge in situ.**

"UPC1" has its uses but in its current form **cannot be taken seriously** as competition for "Freeze Frame".

"ACTION REPLAY" has no tape save, puts 8 files on a disc save and contrary to their claims we have found many programs that it can't handle.

ONLY £39.95

Owners of earlier versions can return them and upgrade for £14.95

SERIOUS WARNING: THIS IS AN EXTREMELY POWERFUL PRODUCT AND IS STRICTLY FOR PERSONAL USE. DON'T COMPARE IT WITH ANY OTHER PRODUCT. NOTHING ELSE OFFERS THE POWER OF "FREEZE FRAME".

ODDS AND ENDS

STAR NL10C The best Commodore ready printer, many features including 120cps draft mode, 27cps near letter quality, tractor and friction feed with semi automatic single sheet feed, most functions front panel switch selectable but perhaps most importantly the **interface cartridge is replaceable.** So if you change to another make of computer you simply purchase a new interface to suit. **ONLY £269.00**

DISC NOTCHER Allows you to easily cut a second write protect notch in your discs so that you can use both sides. A very useful gadget. **ONLY £6.95**

AZIMATE 3000 Lets you realign the head of your data cassette quickly and easily. Supplied with suitable screwdriver. **ONLY £6.95**

3M DISKETTES The best quality discs, available from us at the price of cheap ones. Buy the best, they last a lifetime. Boxes of ten—

SINGLE SIDE/DOUBLE DENSITY	**£11.95**
DOUBLE SIDE/DOUBLE DENSITY	**£13.95**
OR PACKED IN A PLASTIC LIBRARY CASE	
SINGLE SIDE/DOUBLE DENSITY	**£12.95**
DOUBLE SIDE/DOUBLE DENSITY	**£14.95**

COMMODORE COMPATIBLE DATA RECORDER Performs the same operations as the C2N or 1531 but a useful saving. Also includes a pause button (if for use with C16/PLUS4 please state) **ONLY £29.95**

AND THEN THERE WAS DOLPHIN DOS — THE DIFFERENCE IS *STAGGERING*

Like everyone that has had the pleasure of seeing this system in operation you will be amazed by both the speed and ease of use. It is compatible with the majority of commercial software, speeding up both the program loading and SEQ/REL files. Fitting requires the insertion of two sub assembly boards, one in the 1541 and one in the '64/'128. This does not normally entail soldering, although a small amount will be necessary with some machines.

ONLY £69.95

If you require further information please send SAE for fact sheet.

Operates with the CBM 64 or 128 in '64 mode with 1541 disc drive.

THE FEATURES

25x FASTER LOADING (PROGRAM FILES)
12x FASTER SAVING (PROGRAM FILES)
10x FASTER LOADING (SEQUENTIAL FILES)
8x FASTER SAVING (SEQUENTIAL FILES)
3x FASTER LOAD AND SAVE (RELATIVE)
(These figures do not allow for searching)
Easy DOS commands from function keys.
Fast formats 40 tracks giving 749 blocks free.
Centronics driver software inbuilt.
Machine code monitor.
Can be switched out if necessary.

SUBJECT TO AVAILABILITY. ALL ORDERS DESPATCHED THE WORKING DAY AFTER RECEIPT

All prices include P&P and VAT. Send cheque, Postal Order or Credit Card No. Credit Card orders accepted by phone. Callers welcome. Wide range of goods stocked. Trade enquiries welcome. European orders send price as advertised. Outside Europe £2.00 for airmail. Mail order to Evesham please.

EVESHAM MICROS
BRIDGE STREET, EVESHAM, WORCS. WR11 4RY.
Tel: 0386 41989

MICRO CENTRE
1756 PERSHORE ROAD, COTTERIDGE, BIRMINGHAM.
Tel: 021-458 4564

In Australia contact: **MICRO ACCESSORIES of S.A.,** 39b Anderson Walk, Smithfield, South Australia 5114. Tel: (08) 254 6284

DETAILED DIRECTORIES

Get more details from your disks with this routine from L. Jones.

PROGRAM: DETAILED DIR

```
2 PRINT"[CLEAR]":POKE 53280,0
  :POKE 53281,0
4 IF PEEK(49152)<>169 AND PEEK(49153)
  <>8 THEN LOAD"M/CODE",8,1
6 CLR:Z=4
8 DATA "DEL ","SEQ ","PRG ","USR ",
  "REL ","DEL<","SEQ<","PRG<","USR<",
  "REL<"
10 FOR J=0 TO 9:READ T$(J):NEXT
   :GOTO 122
12 OPEN 15,8,15,"I0":INPUT#15,A,B$
   :CLOSE 15
14 IF A<>0 THEN PRINT"[CLEAR,DOWN3,
   RIGHT,WHITE,SPC6]DISK ERROR
   :";"[C7]";A;B$:FOR T=0 TO 2500:NEXT
   :RETURN
16 OPEN 4,Z
18 I$="":N$="":L$=" ":G$="[SPC42]"
20 IF R=1 THEN L$=G$
22 OPEN 1,8,3,"$0"
24 FOR J=1 TO 142:GET#1,A$:NEXT J
26 FOR J=1 TO 16:GET#1,A$
   :IF A$=CHR$(160)GOTO 30
28 N$=N$+A$
30 NEXT J:GET#1,A$,A$
32 FOR J=1 TO 2:GET#1,A$
   :IF A$=CHR$(160)GOTO 36
34 I$=I$+A$
36 NEXT J
38 FOR J=1 TO 92:GET#1,A$:NEXT J
40 PRINT#4,L$;"[CYAN]";" DISK NAME
   :";N$;":ID ";I$;" 2A"
42 IF F=0 THEN PRINT#4,L$;"[C3]";"
    TK:SE:FIL:BLK: PROGRAM[SPC3]NAME
    :S/AD"
44 IF F=1 THEN PRINT#4,L$;"[MAGENTA]";
   " FILE:BLK: PROGRAM[SPC3]NAME :S/AD
   :E/AD"
46 M=M+1:GET#1,K$,T$,S$
   :IF S$=""THEN S$=CHR$(0)
48 F$="":FOR J=1 TO 16:GET#1,A$
   :IF A$=CHR$(160)GOTO 52
50 F$=F$+A$
52 NEXT J
54 FOR J=1 TO 10:GET#1,A$:NEXT J
56 L=0:IF A$<>""THEN L=ASC(A$)
58 GET#1,A$:IF M<8 THEN GET#1,A$,A$
   :GOTO 62
60 M=0
62 SW=ST:IF K$=""GOTO 110
64 K=ASC(K$)-128:IF K>4 THEN K=K-59
66 IF K<1 THEN K=0
68 FL=1:IF K=2 OR K=7 THEN FL=0
70 IF F=1 THEN GOTO 78
72 H$=MID$(STR$(ASC(T$)),2)
   :PRINT#4,L$;"[C1]";LEFT$(G$,
   3-LEN(H$));H$;
74 H$=MID$(STR$(ASC(S$)),2)
   :PRINT#4,"[WHITE]";LEFT$(G$,
   3-LEN(H$));H$;" ";
76 IF F=0 THEN:PRINT#4,"[MAGENTA]";
```

DISK DIRECTORIES ON A C64 CAN only be described as a pain. Unless you have a utility program that will display a directory on your screen the only way of finding what is on your disk is by loading the directory into the computer. Obviously this erases the program that was already in there. Hence the need for a program such as this.

The machine code section of this program (DIRECT.CODE) consists of three small routines.

From C000 to C0A9 is a routine that will display the directory. Once DIRECT.CODE is stored in your C64 then the command SYS 49152 will display the directory of any disk to the screen. Leaving your program intact.

At C0AA to C0E0 is a routine that is used for finding the end address of a program. At C0E1 to C0F9 is a small relocatable routine that produces a striped border effect.

Detailed Dir is a program that was written around these three routines to provide a very useful disk utility.

Firstly you can get a listing of any disk on either the screen or printer. If you are using the printer then you have the option of printing on the right hand side of the paper. As you are no doubt aware directories only fill up half the width of a standard piece of paper. With this program you can feed the paper back into the printer and use the other half.

Secondly the program can also give the following information:
Disk Header.
Track No and Sector No where program is stored.
Number of blocks a program uses.
Program name.
Program start address.
Program end address+1.

If you wish to use any of the machine code routines in your own programs then careful study of Detailed Dir should show you how to use them.

Getting It In

Detailed Dir is a Basic program so you should have no problems typing it in.

DIRECT.CODE is a Basic loader for the machine code. Type this in as a normal program and SAVE it in case you should have made any errors.

Now RUN DIRECT.CODE and if all is well you should be asked to 'PRESS

```
   T$(K);
78 IF F=1 THEN PRINT#4,L$;"[MAGENTA]";
   LEFT$(G$,1);T$(K);
80 IF F=0 THEN H$=MID$(STR$(L),2)
   :PRINT#4,"[GREEN]";LEFT$(G$,
   3-LEN(H$));H$;" ";
82 IF F=1 THEN H$=MID$(STR$(L),2)
   :PRINT#4,"[GREEN]";LEFT$(G$,
   4-LEN(H$));H$;" ";
84 PRINT#4,"[C7]";F$;LEFT$(G$,
   17-LEN(F$));"[YELLOW]";
86 IF K=0 GOTO 108
88 IF K=2 OR K=7 OR F=1 THEN OPEN 2,8,
   4,"0:"+F$+","+T$(K)+",R"
90 A=0:IF FL=1 THEN 98
92 GET#2,A$,B$:A=0:IF A$<>""THEN A=AS
   C(A$)
94 B=0:IF B$<>""THEN B=ASC(B$)
96 GOSUB 114
98 IF F=0 THEN 106
100 POKE 785,170:POKE 786,192
   :A=A+USR(0)
102 IF FL=1 THEN PRINT#4,A;"BYTES";
   :GOTO 106
104 PRINT#4,"[C3]";" ";:A%=A/256
   :A=A-A%*256:B=B+A%:GOSUB 114
106 CLOSE 2
108 PRINT#4
110 IF SW=0 GOTO 46
112 CLOSE 1:CLOSE 4:GOTO 120
114 X=B/16:GOSUB 116:X=A/16
116 FOR J=1 TO 2:X%=X:X=(X-X%)*16
   :IF X%>9 THEN X%=X%+7
118 PRINT#4,CHR$(X%+48);:NEXT J:RETURN
120 RETURN
122 PRINT"[CLEAR]";:E$="[MAGENTA,
   RVSON,SV,RVSOFF]":GOSUB 186
124 PRINT"[HOME,DOWN3,RIGHT2,WHITE]
   THIS[SPC,C7]PROGRAM[SPC,WHITE]WILL
    ALLOW YOU TO[SPC,C7]PRINT"
126 PRINT"[DOWN,RIGHT4,WHITE]
   YOUR DISK DIRECTORY TO THE[SPC,C7]
   SCREEN[WHITE]":PRINT"[DOWN,RIGHT2,
   C7]OR PRINTER."
128 PRINT"[DOWN,RIGHT3,WHITE]YOU WILL
    ALSO HAVE A CHOICE OF THE"
130 PRINT"[DOWN,RIGHT3]FOLLOWING
   :-[C7]END[WHITE,SPC]
   OF PROGRAM ADDRESS,"
132 PRINT"[DOWN,RIGHT6]PRINT[C7,SPC]
   LEFT[WHITE,SPC]OR[C7,SPC]RIGHT
   [WHITE,SPC]OF PAPER,"
134 PRINT"[DOWN,RIGHT3 C7]CHECK[SPC,
   WHITE]YOUR DISK BEFORE YOU DUMP
    TO"
136 PRINT"[DOWN,RIGHT3]
   SCREEN OR PRINTER WITHOUT LOOSING"
138 PRINT TAB(14)"[DOWN]THIS PROGRAM."
   :GOSUB 190
140 E$="[GREEN,RVSON,SV,RVSOFF]"
   :PRINT"[CLEAR]";:GOSUB 186
   :PRINT"[HOME,DOWN2,WHITE,RIGHT,
   SPC6]FUNCTIONS AT YOUR DISPOSAL"
142 PRINT"[UP,RED,RIGHT,SPC6,CT27]"
144 PRINT TAB(11)"[C7]<<<[MAGENTA,SPC]
   SCREEN DUMP[C7,SPC]>>>"
146 PRINT TAB(13)"[UP,RIGHT2,RED,CT11]
   ":PRINT"[DOWN,RIGHT,YELLOW,SPC2]
   CHECK[SPC,CYAN]FOR CORRECT DISK"
148 PRINT"[DOWN2,RIGHT,YELLOW,SPC2]
   PRINT[SPC,CYAN]TO SCREEN[WHITE]-
   [CYAN]END ADDRESS"
150 PRINT"[DOWN,RIGHT,YELLOW,SPC2]
   PRINT[SPC,CYAN]TO SCREEN[WHITE]+
   [CYAN]END ADDRESS"
152 PRINT"[DOWN2,RIGHT,YELLOW,SPC2]
   NEXT MENU -->>>":GOSUB 154:GOTO 166
154 A1$="[HOME,DOWN6]":A2$="[RIGHT31]"
   :A3$="[C7,SU]────[SI]"
156 A4$="[UP]":A5$="[SJ,SC4,SK]"
   :AA$=A2$+A3$:AB$=A2$+A5$
158 F1$="[SB,WHITE,SPC]F1[C7,SPC,SB]"
   :F3$="[S-,WHITE,SPC]F3[C7,SPC,S-]"
   :F5$="[SB,WHITE,SPC]F5[C7,SPC,SB]"
   :F7$="[SB,WHITE,SPC]F7[C7,SPC,SB]"
   :F8$="[SB,WHITE,SPC]F8[C7,SPC,SB]"
160 PRINT A1$+AA$:PRINT A2$+A4$+F1$
   :PRINT AB$:PRINT A4$+AA$
   :PRINT A2$+F3$
162 PRINT A4$+AB$:PRINT AA$
   :PRINT A2$+A4$+F5$:PRINT AB$
   :PRINT A4$+AA$:PRINT A2$+F7$
164 PRINT A4$+AB$:RETURN
166 GET A$:IF A$="[F1]"THEN PRINT"
   [CLEAR]";:GOSUB 186:GOTO 176
168 IF A$="[F3]"THEN PRINT"[CLEAR]"
   :Z=3:GOSUB 254:GOSUB 190:GOTO 140
170 IF A$="[F5]"THEN PRINT"[CLEAR]"
   :Z=3:GOSUB 256:GOSUB 190:GOTO 140
172 IF A$="[F7]"THEN GOTO 200
174 SYS 49377:GOTO 166:REM GOTO166
   :SYS49377
176 OPEN 15,8,15,"I0":INPUT#15,A,B$
   :CLOSE 15
178 IF A<>0 THEN PRINT"[DOWN3,RIGHT,
   WHITE,SPC6]DISK ERROR
   :";"[YELLOW]";A;B$:FOR T=0 TO 2500
   :NEXT:GOTO 140
180 PRINT"[DOWN3,RIGHT3,C6,SPC2]USE
    SPACE BAR TO PAUSE LISTING"
182 PRINT TAB(8)"[DOWN,RIGHT3,C3]ANY
    KEY TO RESUME.":FOR DE=1 TO 2000
   :NEXT:SYS 49152
184 GOSUB 190:GOTO 140
186 FOR X=1 TO 39:PRINT E$;:NEXT
   :FOR X=1 TO 23:PRINT E$;"[DOWN,
   LEFT]";:NEXT
188 FOR X=1 TO 39:PRINT E$;"[LEFT2]";
   :NEXT:FOR X=1 TO 23
   :PRINT E$;"[UP,LEFT]";:NEXT:RETURN
190 PRINT"[HOME,DOWN23,RIGHT7,YELLOW]
   PRESS ANY KEY TO CONTINUE"
192 FOR DE=1 TO 200:NEXT:SYS 49377
194 PRINT"[UP,RIGHT7,RVSON,YELLOW]
   PRESS ANY KEY TO CONTINUE[RVSOFF]"
   :FOR DE=1 TO 200:NEXT
```

```
196 GET A$:IF A$=""THEN 190
198 RETURN
200 E$="[CYAN,RVSON,SV,RVSOFF]"
   :PRINT"[CLEAR]";:GOSUB 186
202 PRINT"[HOME,DOWN2,WHITE,RIGHT,
   SPC6]FUNCTIONS AT YOUR DISPOSAL"
204 PRINT"[UP,RED,RIGHT,SPC6,CT27]"
206 PRINT TAB(10)"[YELLOW]<<<[MAGENTA,
   SPC]PRINTER DUMP[YELLOW,SPC]>>>"
208 PRINT TAB(10)"[RIGHT4,RED,UP,CT12]
   "
210 PRINT"[DOWN,RIGHT,YELLOW,SPC2]
   PRINT LEFT[SPC,CYAN,SPC]
   -END ADDRESS"
212 PRINT"[DOWN2,RIGHT,YELLOW,SPC2]
   PRINT LEFT[SPC,WHITE,SPC]+[CYAN]
   END ADDRESS"
214 PRINT"[DOWN,RIGHT,YELLOW,SPC2]
   PRINT RIGHT[SPC,WHITE]-[CYAN]END
    ADDRESS"
216 PRINT"[DOWN2,RIGHT,YELLOW,SPC2]
   PRINT RIGHT[SPC,WHITE]+[CYAN]END
    ADDRESS"
218 PRINT"[DOWN,RIGHT,YELLOW,SPC2]
   CHANGE DEVICE NUMBER NOW[WHITE,SPC,
   LEFT]";Z
220 GOSUB 154:PRINT AA$
   :PRINT A2$+A4$+F8$:PRINT AB$
   :GOTO 222
222 GET A$:IF A$="[F1]"THEN R=0
   :GOSUB 254:GOTO 200
224 IF A$="[F3]"THEN R=0:GOSUB 256
   :GOTO 200
226 IF A$="[F5]"THEN R=1:GOSUB 254
   :GOTO 200
228 IF A$="[F7]"THEN R=1:GOSUB 256
   :GOTO 200
230 IF A$="[F8]"THEN PRINT"[CLEAR]";
   :GOSUB 186:GOTO 236
232 IF A$=" "THEN 2
234 GOSUB 248:GOTO 222
236 PRINT"[HOME,DOWN3,RIGHT2,WHITE,
   SPC7]INPUT A VALUE[SPC,MAGENTA](
   [GREEN]0-255[MAGENTA])"
238 PRINT"[DOWN,RIGHT3,SPC5]DO NOT
    USE A VALUE OF 3"
240 PRINT"[DOWN,RIGHT3,SPC4]AS THIS
    IS FOR THE SCREEN"
242 OPEN 1,0:PRINT"[C7,DOWN2,RIGHT6,
   SPC2]INPUT DEVICE NUMBER ? 4[SPC2,
   LEFT3]";:INPUT#1,Z$
244 PRINT:CLOSE 1:Z=VAL(Z$)
   :IF Z<0 OR Z>255 OR Z=3 THEN PRINT
   "[UP3]":GOTO 236
246 GOSUB 190:GOTO 200
248 PRINT"[HOME,DOWN23,RIGHT5,YELLOW]
   PRESS SPACE TO RESTART PROGRAM[UP]"
250 FOR DE=1 TO 200:NEXT:SYS 49377
252 PRINT"[UP,RIGHT5,RVSON,YELLOW]
   PRESS SPACE TO RESTART PROGRAM[UP2]
   ":FOR DE=1 TO 200:NEXT:RETURN
254 F=0:GOSUB 12:RETURN
256 F=1:GOSUB 12:RETURN
```

SPACE TO SAVE'. If you have made any errors in the program then correct them before trying again.

When you press space the program "M/CODE" will be SAVEd on to your disk. This is the program that Detailed Dir will look for when you RUN it.

If you want to use the machine code in your own programs then "M/CODE" should be loaded with the extension,8,1 so that it loads at memory location 49152 ($C000) onwards.

```
PROGRAM:  DIRECT.CODE

2000 FOR L=0 TO 15:CX=0:FOR D=0 TO 15
   :READ A:CX=CX+A:POKE 49152+L*16+D,A
   :NEXT D
2010 READ A:IF A<>CX THEN PRINT"ERROR
    IN LINE";2040+(L*10):STOP
2020 NEXT L
2040 DATA 169,8,133,75,169,0,133,104,
   169,128,133,105,32,68,229,169,1824
2050 DATA 36,133,2,169,1,133,183,133,
   184,169,96,162,8,133,185,134,1861
2060 DATA 186,169,2,162,0,133,187,134,
   188,32,213,243,165,75,32,9,1930
2070 DATA 237,165,185,32,199,237,169,
   0,133,144,160,3,132,183,32,19,2030
2080 DATA 238,133,195,32,19,238,133,
   196,164,144,208,87,164,183,136,208,
   2478
2090 DATA 235,162,6,169,32,32,210,255,
   202,208,250,169,1,141,134,2,2208
2100 DATA 166,195,165,196,32,205,189,
   169,14,141,134,2,169,32,32,22,1863
2110 DATA 231,32,19,238,166,144,208,
   43,201,0,240,6,32,22,231,76,1889
2120 DATA 113,192,32,237,246,240,28,
   32,228,255,240,13,201,32,208,9,2306
2130 DATA 32,228,255,240,251,201,3,
   240,10,169,13,32,22,231,160,2,2089
2140 DATA 76,60,192,32,66,246,32,239,
   237,96,169,0,162,4,149,98,1858
2150 DATA 202,16,251,169,160,133,97,
   162,2,32,198,255,230,101,208,10,
   2226
2160 DATA 230,100,208,6,230,99,208,2,
   230,98,32,228,255,165,144,240,2475
2170 DATA 235,32,204,255,198,97,6,101,
   38,100,38,99,38,98,16,244,1799
2180 DATA 96,162,15,142,32,208,160,73,
   136,208,253,202,234,234,165,197,
   2517
2190 DATA 201,64,240,239,169,240,141,
   32,208,96,0,0,0,0,255,255,2140
3000 REM ** READY TO SAVE **
3010 PRINT "[CLEAR,DOWN3,SPC4]DATA
    ALL CORRECT"
3020 PRINT "[DOWN4,SPC]PRESS[SPC,
   RVSON]SPACE[RVSOFF,SPC]TO SAVE"
3030 GET K$:IF K$<>" " THEN 3030
3040 POKE 43,0:POKE 44,192:POKE 45,253
   :POKE 46,192:CLR
3050 SAVE "M/CODE",8,1
```

Screen splitting antics from Joe Nicholson. For your C-16.

Programming The C-16

IN THIS ARTICLE I SHALL BE concentrating on setting up and using split screens. These are extremely useful in a wide range of games and have the distinct advantage of making available extra memory within the machine and are based on raster interrupts.

Raster Interrupts

The one type of interrupt not discussed last month was the raster interrupt. In order to explain how to use raster interrupts I had better first explain what they are! Rasters basically relate to the screen picture sent by the C-16 to the TV. As the computer outputs the screen signals to the television it scans from the top of the screen to the bottom, 50 times a second. Therefore each pixel line that is generated (there are eight pixel lines in each character line of text) is called a raster line. There are, therefore, 25×8=200 raster lines for a screen. The border uses a further 111 lines at the top and bottom of the screen display for the PAL colour television system as used in the UK. On the American NTSC system there are only 61 raster lines for the border.

The two memory locations $FF28 and $FF1C (65308 and 65309 decimal) are the vertical raster count registers. These two bytes can be read to find which raster line is currently being displayed. As the picture is displayed from top to bottom, these registers increment from zero to 311 (or 261 for the NTSC system), before going back to zero again for the next frame.

Bit zero of address $FF1C (65308 decimal) contains the highest bit of the raster compare register and address $FF1D (65309) contains the lower eight bits. Also, address $FF1E (65310 decimal) contains the upper eight bits of the nine bit horizontal raster position register. This increments so fast that its only real use to the programmer is to generate random numbers.

Figure 1 shows a short machine code program to demonstrate the basics behind operating a split screen. The loop in lines 10100-10120 waits for the raster register to equal 123 decimal, i.e. just below halfway down the screen. Lines 10100-10170 then set the colour of the background border to cyan. The loop in lines 10200-10220 waits for the raster scan to equal zero again, i.e. the very top of the screen. Lines 10250-10270 then set the background and border to white — it then goes round again. The result is a two colour screen. Figure 2 shows this listing as code in case you don't have my C-16 Assembler which was published in the June 1985 edition of Your Commodore.

Right, so now we have looked at what rasters are, let's start thinking about raster interrupts.

Addresses $FF0A and $FF0B (65290 and -65291 decimal) contain the nine bits of the raster compare register. Address $FF0B holds the lower eight bits, and bit zero of address $FF0A holds the most significant bit. The remaining bits of $FF0A hold the interrupt mask register, so be careful not to alter any of these when changing zero.

When the raster line count in registers $FF1C and $FF10 equals the value in the raster compare register, bit one of the interrupt status register at $FF09 is set (see my interrupts article). If bit zero of the interrupt enable register ($FF0A) is also set, an interrupt is generated. As explained in my previous article addresses $0314 and $0315 (88 and 89 decimal) hold the address of the interrupt vector which is usually $CE0E, but can be altered to go to a user routine.

Therefore by setting bit one of the interrupt mask register, setting the raster compare registers to the line where you wish to interrupt, and redirecting the interrupt vector you should get an interrupt — right? Well, it's not so simple unfortunately, because the C-16 also uses the raster interrupt itself all the time for its own split screen routines for graphics modes two and four. Unfortunately these routines are also active in all the other graphics modes and can never be turned off.

So why not change the machine's existing split screen routines to create your own interrupts and split screens? Well, you can't. In fact you can't even change the line at which the screen splits. Therefore to create your own raster interrupts, split screens etc., you have to write it all yourself, including some of the interrupt service routines, because these also mess about with the split screen.

So is it all worth the effort? The answer is 'yes'. Split screens are more useful for the C-16 than just about any other computer due to its limited memory capacity., At present, when the high-resolution mode is selected, a huge 10.3 Kbyte portion of the 12.3 Kbyte maximum available memory is used up leaving you with a couple of thousand bytes in which you can do very little. However if you split the screen and use say 15 of the 25 available high-resolution character lines, leaving the remaining 10 in low-resolution mode, you can save 3200 bytes in the high-res screen space plus another 400 bytes in the luminance and 400 bytes in the chrominance tables. We also gain another 600 bytes in the low-res screen and another 600 bytes in the low-res colours giving a total of 3200+400+400+600+600=5200 bytes saved. Thus we have a total of 7245 bytes free compared with the original 2045. Admittedly this memory is scattered all over the place, but this is not a serious problem for machine code programs. There will be much more about using the additional space in a future article. For now just bear in mind that it is possible to have three and a half times as much memory when in hi-res mode.

Split Screen Routine

Figure 3 shows the assembly listing for the split screen routine. Figure 4 shows the code for the split screen. Enter Figure 4 instead of Figure 3 if you don't have the C-16 Assembler. In Figure 4 lines 10000-10020 load the machine code. The routine is stored in an unused area of memory at $0600 (1536 decimal) and is around 200 bytes long.

I have included a table which is a breakdown of the system variables used in the split screen program:

Table

Address	Description
$D0	position in line
$D1	the number of splits × 2 (e.g. for 3 splits this byte is 6)
$D2	raster line number for 1st split
$D3	data byte for 1st split
$D4	raster line number for 2nd split
$D5	data byte for 2nd split
$D6	raster line number for 3rd split
$D7	date byte for 3rd split
$D8	raster line number for 4th split
$D9	data byte for 4th split
$DA	working byte for split routine.

Note that the position of the splits must follow in order down the screen, i.e. the raster line for the second split must be greater than that for the first split. The raster lines for the screen start at one at the top of the screen and go down to 202 at the bottom. Although the screen has only 200 raster lines and therefore in theory the last line of the screen should be (200+1$, for some reason the lines end at 202. Also, if you need an interrupt right at the top of the screen it is better to set the raster line to zero instead of one as the change will take place off the screen avoiding any flicker. To avoid flicker and attribute difficulties, the raster line number for the split should be set from the following equation:

Raster line no. = (Character line)+8 + 1

Note that the character line can be from zero to 24. The equation makes the screen split at the bottom of the character line; flicker can occur if the raster line is set to the middle of the character line.

Assembly Listing

Lines 10100-10290 — turn on the split screen, altering the position of the interrupt to $0630. They also 'enable' the raster interrupts and sets the raster compare register up to the first split.
Lines 10300-10380 — turn off the split screen system by redirecting the interrupt to the ROM service routine.
Lines 11000-11520 — contain the interrupt service routine. This routine, mainly adapted from the C-16's ROM, contains all the code necessary to make it read the keyboard, update the clock, etc. Note that it is not possible in this case to use the C-16's own ROM routines because these would ruin the split screen.
Lines 11050-11080 — perform a split if the interrupt was a raster interrupt.
Lines 11100-11170 — deal with all the RS232 interrupts used in operating printers, disk drives, etc.
Lines 11170-11280 — turn off the 'raster interrupt' flag in the raster status register if there was a raster interrupt and also set the new split line position in the raster compare register.
Lines 11290-11310 — decide whether it is a new 1/50th of a second and if it is (i.e. if the next split is the top split), update the clock, read the keyboard and increment sound duration registers in the routine at lines 11400-11520. If it is not however, they jump to the IRQ 'exit' ROM routine in line 11350.
Lines 12000-12090 — perform a split. This routine interrupts the Data byte into its appropriate instructions. Because all screen changes must be done at once, it pushes the numbers to be changed into registers X and Y and stores all the values toward the end of the routine, at lines 12500-12560.
Lines 12100-12110 — fetch the data byte.
Lines 12120-12180 — deal with the 'TED fetches from RAM/ROM' bit, altering bit 2 of address $FF12.
Lines 12182-12194 — deal with the 'position of the video matrix' bit by setting byte $FF15 to 8 for low-res or with the position of the start of the luminance table (stored in address $07FB for a high-res screen).
Lines 12200-12260 — deal with the '24/25 line screen', the 'bit-map' mode and the 'extended background colour' mode bits by altering bits 3,5 and 6 of register $FF06 accordingly.
Lines 12300-12390 — deal with the '38/40 column screen' bit by altering bits 3 and 4 of register $FF07 accordingly.
Lines 12500-12560 — store all these values virtually simultaneously to reduce flicker between splits.
Lines 12600-12700 — move the split position counter on 1 to the next split, or back to the beginning if the end of the split table is reached.

Data Bytes

The data byte for each split contains all the information about the window below it. For example if you want a screen split in the middle with the top half in high-res and the bottom in low-res, set the date byte for a split at the top of the screen for high-res, and the data byte for the a split half way down for low-res.

Here is how the bits of the data byte are arranged:
Bit 0: 38/40 Column Screen: 0 for 38 columns and 1 for 40 columns. This is used in smooth scrolling and will be explained in a later article.
Bit 1: Multicolour Mode: 0 for mode OFF and 1 for mode ON.
Bit 2: TED Fetches from ROM/RAM. In low-res normal mode and character set is contained in ROM, so TED (the text editor device in the C-16) therefore fetches information from ROM. In programmable character mode (see my article in the November 1985 issue) the character set is in RAM, so TED therefore fetches from RAM. In high-res mode also TED fetches from RAM. The bit is 0 for 'TED fetchs from RAM' and 1 for 'TED fetches from ROM'.
Bit 3: 24/25 Line Screen. This is also used in smooth scrolling. 24 line mode cuts off half a character line from the top and half a character line from the bottom of the screen (the actual visible effect depends on where the window below the split is. If the window is in the middle of the screen only, this bit will have no effect at all).
The bit is 0 for 24 lines and 1 for 25 lines.
Bit 4: not used.
Bit 5: Bit Mapped (high-res mode). The bit is 0 for OFF (for low-res screen) and 1 for ON (for high-res screen).
Bit 6: Extended Background Colour Mode (see my article in the March 1986 issue of Your Commodore). The bit is 0 for OFF and 1 for ON.
Bit 7: Position of Video Matrix. 0 sets the video matrix address to 2048 for low-res mode. 1 sets the address to the start of the luminance table for high-res mode. Therefore the bit is 0 for low-res and 1 for high-res.

For example to set a normal low-res screen bits 0,2 and 3 should be set, the rest of the bits should be reset. Therefore the data byte would be 1+4+8=13=$OD. For a standard high-res screen (non-multicolour) the byte should be 128+32+8+1=169 =$A9.

Operating Instructions

To turn on the split screen routine type: SYS 1536.
To turn off the split screen routine type: SYS 1568.

Then to set up a simple split screen with the top half of the screen in high-res and the bottom half in low-res:

POKE 209,4 — for (2 splits × 2)
POKE 210,0 — split at top of the screen
POKE 211,169 — data byte for 1st high-res split
POKE 212,97 — for a split at the 12th line
POKE 213,13 — data byte for 2nd split (low-res)
SYS 1536 — turn it ON.

The GRAPHIC commands do not have an effect any more, but serve their purpose in specifying which window we are using in Basic. E.g. GRAPHIC 0 specifies low-res, GRAPHIC 1 specifies high-res and GRAPHIC 3 specifies high-res multicolour mode. Also to clear a particular screen put a one after the end of the GRAPHIC command. For instance, if you have entered the above commands in you should now see rubbish on the top half of the screen. Type GRAPHIC 1,1 and the top high-res window will clear. Note that either GRAPHIC 2,1 or GRAPHIC 4,1 clears both the low and the

high resolution screens simultaneously.

Now type CIRCLE,150,45, 45 (don't forget the initial comma) to draw a circle in the high-res window.

Then type PAINT,150,45, to fill the circle in.

Finally type SYS 1568 to get back to normal.

The above example should have given you some ideas on how to use split screens, but does not convey much of the power of these routines. Figure 5 lists a demonstration program which is designed to show up to four screens in action.

Datamaka

The following routine, although not having much to do with split screens, is an important utility when dealing with machine code and data. This routine was used when I printed the code in this and the previous two installments of Programming the C-16. Basically it turns a given block of code into DATA statements with the option of a checksum at the end of each line. This program greatly reduces the amount of time you spend turning your machine code games etc. into data so that you can send them to Your Commodore!

Figure 6 shows the DATAMAKA program. When the program is RUN enter the START ADDRESS and END ADDRESS of the code in response to the questions. Next enter the starting Basic line number of the block of DATA statements in response to LINE NUMBER ? and the step between line numbers in response to STEP FOR LINE NUMBERS ? Lastly enter "Y" or "N" in response to CHECKSUM(Y/N)? depending on whether you want a checksum (sum of all the numbers in a line and a useful check) at the end of each line. The program then goes off and constructs the DATA statements. Note that it ends the last line with a '-1'.

In setting up the DATA statements one or two tricks are used, notably the automatic execution of the lines putting them into memory once every five data statements have been displayed on the screen. This is done by POKEing characters into the keyboard buffer so that, when the end of line 15100 is reached, commands are stored in the keyboard buffer to make it enter these lines into memory. The actual characters are a HOME directing the cursor to the top of the screen, followed by six ENTER instructions. The number of characters in the keyboard buffer is POKEd into the 'index to keyboard queue' register at address 239 in line 15100.

The other interesting technique employed is the way the program is restarted automatically to continue DATAMAKAing. Every time a new DATA line is ENTERed into memory the computer wipes all the variables, so all the information on end addresses etc. is lost. To overcome this a line is printed on the screen (by line 15030) that records all the variables in the form of commands, for instance, S=15020:P=1000: and so on at the end of each group of five DATA lines followed by a GOTO 15010. When the C-16 executes this line all the variables are restored and the program continues execution.

The variable A in line 1000 should be set to the start address of the code. Line 10020 checks the sum of the numbers in each line with the checksum value and if they do not agree the program exits with 'DATA ERROR IN' followed by the offending line number. This program line should be changed to 10020 LOOP and line 10010 should change to

10010 POKE A,D:A=A+1

if the checksum mode was not specified when the data was generated. To put this program at the start of the data and delete the rest of DATAMAKA delete line 500 then type DELETE 15000-15100.

Well, that concludes this month's article, I hope it provided some useful routines and some food for thought. Next time I shall be applying this and other routines to creating extra memory on the C-16 and showing how to have 4K for Basic in high-res mode and still have untouched high-res and low-res screens. Impossible? Reserve your copy at the newsagents now!

PROGRAM: FIGURE 1

```
START: 10000  ORG $0600
 10100  :L1 LDA $FF1D
 10110  CMP #123
 10120  BNE R:L1
 10150  LDA #$63
 10160  STA $FF15
 10170  STA $FF19
 10200  :L2 LDA $FF1D
 10210  CMP #0
 10220  BNE R:L2
 10250  LDA #$71
 10260  STA $FF15
 10270  STA $FF19
 10300  JMP :L1
>> OK.
```

PROGRAM: FIGURE 2

```
10000 A=1536:DO:B=0:FORC=1TO
16:READD:IFD=-1THENPRINT"OK.
.":END
10010 B=B+D:POKEA,D:A=A+1:NE
XT:READD
10020 IFD<>BTHENPRINT"DATA E
RROR IN LINE";PEEK(63)+256*P
EEK(64):END:ELSELOOP
20000 DATA 173,29,255,201,12
3,208,249,169,99,141,21,255,
141,25,255,173, 2517
20010 DATA 29,255,201,0,208,
249,169,113,141,21,255,141,2
5,255,76,0, 2138
20020 DATA 6,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0, 6,-1
```

PROGRAM: FIGURE 3

```
START: 10000  ;SPLIT SCREEN 16
 10010  ;(C)1986 JOE NICHOLSON
 10020  ORG $0600
 10100  ;
 10110  ;ON SPLIT SCREEN
 10120  :ON SEI
 10130  LDA #$30
 10140  STA $0314
 10150  LDA #$06
 10160  STA $0315
 10200  LDA #1
 10210  STA $D0
 10220  LDA $D2
 10230  STA $FF0B
 10240  LDA $FF0A
 10250  AND #$FE
 10260  ORA #$02
10270  STA $FF0A
10280  CLI
10290  RTS
10300  ;
10310  ;OFF SPLIT SCREEN
10320  :OF SEI
10330  LDA #$0E
10340  STA $0314
10350  LDA #$CE
10360  STA $0315
10370  CLI
10380  RTS
11000  ;
11010  ;INTERRUPT SERVICE ROUTINE
11020  ORG $0630
11050  LDA $FF09
11060  AND #$02
11070  BNE R:RA
11100  :NR BIT $07D8
11110  BPL R:EA
11120  LDA $FD01
11130  STA $07D4
11140  BPL R:EA
11150  JSR $EA95
11160  JSR $EA5B
11170  :EA JSR $E3E4
11200  LDA $FF09
11210  AND #2
11220  BEQ R:EX
11230  STA $FF09
11250  LDX $D0
11260  LDA $D2,X
11270  INC $D0
11280  STA $FF0B
11290  LDA $D0
11300  CMP #1
11310  BEQ R:CL
11350  :EX JMP $FCBE
11400  :CL JSR $CFBF
11410  JSR $CECD
11420  LDA $FB
11430  PHA
11440  LDA #0
11450  STA $FB
11460  PHA
11470  CLI
11480  JSR $DB11
11490  PLP
11500  PLA
11510  STA $FB
11520  JMP $FCBE
12000  ;
12010  ;PERFORM SPLIT
12100  :RA LDX $D0
12110  LDA $D2,X
12120  STA $DB
12130  AND #4
12140  STA $DA
12150  LDA $FF12
12160  AND #251
12170  ORA $DA
12180  TAX
```

```
12182  LDA $DB
12184  AND #128
12186  BNE R:HI
12188  LDA #8
12190  BNE R:SU
12192  :HI LDA $07FB
12194  :SU STA $FF14
12200  LDA $DB
12210  AND #104
12220  STA $DA
12230  LDA $FF06
12240  AND #151
12250  ORA $DA
12260  TAY
12300  LDA $DB
12310  AND #3
12320  ASL A
12330  ASL A
12340  ASL A
12350  STA $DA
12360  LDA $FF07
12370  AND #231
12380  ORA $DA
12520  STX $FF12
12530  STY $FF06
12540  STA $FF07
12600  INC $D0
12610  LDA $D0
12620  CMP $D1
12630  BNE R:NB
12650  LDA #0
12660  STA $D0
12700  :NB JMP :NR
```

PROGRAM: FIGURE 4

```
10000 A=1536:D0:B=0:FORC=1TO16:READD:IFD=-1THENPRINT"OK.":END
10010 B=B+D:POKEA,D:A=A+1:NEXT:READD
10020 IFD<>BTHENPRINT"DATA ERROR IN LINE";PEEK(63)+256*PEEK(64):END:ELSELOOP
20000 DATA 120,169,48,141,20,3,169,6,141,21,3,169,1,133,208,165, 1517
20010 DATA 210,141,11,255,173,10,255,41,254,9,2,141,10,255,88,96, 1951
20020 DATA 120,169,14,141,20,3,169,206,141,21,3,88,96,0,255,0, 1446
20030 DATA 173,9,255,41,2,208,75,44,216,7,16,14,173,1,253,141, 1628
20040 DATA 212,7,16,6,32,149,234,32,91,234,32,228,227,173,9,255, 1937
20050 DATA 41,2,240,18,141,9,255,166,208,181,210,230,208,141,11,255, 2316
20060 DATA 165,208,201,1,240,3,76,190,252,32,191,207,32,205,206,165, 2374
20070 DATA 251,72,169,0,133,251,72,88,32,17,219,40,104,133,251,76, 1908
20080 DATA 190,252,166,208,181,210,133,219,41,4,133,218,173,18,255,41, 2442
20090 DATA 251,5,218,170,165,219,41,128,208,4,169,8,208,3,173,251, 2221
20100 DATA 7,141,20,255,165,219,41,104,133,218,173,6,255,41,151,5, 1934
20110 DATA 218,168,165,219,41,3,10,10,10,133,218,173,7,255,41,231, 1902
20120 DATA 5,218,142,18,255,140,6,255,141,7,255,230,208,165,208,197, 2450
20130 DATA 209,208,4,169,0,133,208,76,55,6,255,0,255,0,255,0, 1833,-1
```

PROGRAM: FIGURE 5

```
5 GOTO1000
10 COLOR4,1:COLOR1,2,7:COLOR0,1:DRAW,3,3TO148,3TO148,92TO3,92TO3,3
17 CIRCLE,148,3,30,,180,270:PAINT,140,4
20 COLOR2,5,5:COLOR3,5,6
30 DRAW2,55,50TO10,60TO50,40TO60,30TO65,28:DRAW3,65,28TO70,27TO80,31
40 DRAW3,80,31TO118,22TO85,40TO87,75TO55,50
50 DRAW2,87,75TO65,28
60 PAINT2,20,56,1:PAINT3,70,30,1:RETURN
1000 POKE209,2:POKE210,0:POKE211,13:SYS1536
1005 COLOR0,1:COLOR1,2:COLOR4,7,4
1010 GRAPHIC4,1:PRINT"(HOME)SETTING UP ONLY ONE SPLIT:":LIST1000
1030 PRINT"IS EQUAL TO NO SPLIT SCREEN AT ALL.":GETKEYA$
1050 PRINT"(CLR)":LIST1100:PRINT"SETS UP A 2ND SPLIT:":COLOR0,6,5:GRAPHIC1,1
1100 SYS1568:POKE209,4:POKE212,57:POKE213,169:SYS1536
1110 CHAR,9,11,"THIS TIME IN HIGH RES."
1120 FORA=1TO20STEP2:BOX,64-A,80-A,256+A,104+A,B,B:NEXT:GETKEYA$
1200 PRINT"(CLR)":LIST1250:PRINT"SETS THE 3RD SPLIT"
1250 SYS1568:POKE209,6:POKE214,129:POKE215,77:SYS1536
1290 COLOR3,5,5:COLOR0,1
1300 POKE65304,98:PRINT"(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)THIS (RVS)╮●(OFF) ╲╱ EXTENDED (RVS)|♠‾╯|_┌╱╱‾(OFF) ┤┌└┌╱_"
1310 PRINTSPC(18)"╲┌‾‾":GETKEYA$
1400 PRINT"(HOME)FINALLY":LIST1450:PRINT"SETS THE 4TH SPLIT TO MULTI COLOUR"
1410 PRINT"HI-RES MODE:"
1450 SYS1568:POKE209,8:POKE216,161:POKE217,171:SYS1536
1455 GRAPHIC3:FORA=20TO24:CHAR,0,A,"                                        ":NEXT
1460 COLOR2,7,4:COLOR1,8,6:FORA=1TO40:B=INT(RND(0)*360)
1470 Y=INT(RND(0)*40)+161:X=INT(RND(0)*150):BOXINT(RND(0)*3)+1,X,Y,X+10,Y+10,B,1:NEXT
1480 GETKEYA$:FORA=1TO16:FORB=0TO7:COLOR3,A,B:NEXT:NEXT
2000 COLOR4,1:COLOR1,7:GRAPHIC4,1
2010 SYS1568:POKE209,2:POKE210,0:POKE211,13:SYS1536
2015 PRINT"(HOME)(DOWN)(DOWN)(DOWN)(DOWN)(RGHT)(RGHT)(RGHT)SETTING UP THE GALAXY...(DOWN)(DOWN)(LEFT)(LEFT)(LEFT)(LEFT)PLEASE WAIT.":GOSUB10
2020 SYS1568:POKE209,4:POKE211,171:POKE212,97:POKE213,13:SYS1536
2050 PRINT"(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)CONDITION (RED)RED(WHT)","STARDATE:243.2"
2060 PRINT"ENERGY:42346","QUADRANT:10,17"
2070 PRINT"SHIELDS:500"
2100 PRINT"(DOWN)(RED)A MUTANT KLINGON HAS ENTERED THIS         QUADRANT!!"
2110 PRINT"(WHT)SCOTTIE SAYS THAT IF WE DON'T GET SOME DILITHIUM CRYSTALS SOON THE SHIP IS IN"
2120 PRINT"IMMINENT DANGER OF EXPLODING!"
2150 VOL7:FORA=1TO6:FORB=500TO1000STEP10:SOUND1,B,1:SOUND2,1000-B,1:NEXT:NEXT
2160 GRAPHIC0:FORA=16TO23:CHAR,0,A,"                                        ":NEXT
2170 INPUT"(UP)(UP)(UP)(UP)(UP)(UP)(UP)WHAT ARE YOUR ORDERS CAPTAIN";A$
2180 IFA$<>"FIRE"ANDA$<>"TORPEDO"ANDA$<>"SHOOT"THENGOTO2160
2200 PRINT"FIRE WHEN READY";:GETKEYA$
2210 PRINT" (*130)(RVS)FIRE(OFF)(*132)":FORA=1023TO700STEP-1:SOUND1,A,1:SOUND2,A-100,1:NEXT
2220 SOUND3,500,200:PRINT"(YEL)A HIT!!(WHT)":FORA=7TO0STEP-1:FORB=16TO2STEP-1
2230 COLOR3,B,A:NEXT:NEXT:GETKEYA$:COLOR3,5,6
```

PROGRAM: FIGURE 6

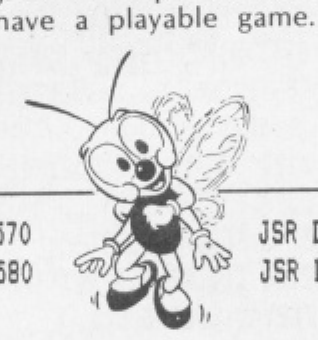
```
5000 GOTO15000
10000 A=4096:D0:B=0:FORC=1TO16:READD:IFD=-1THENPRINT"OK.":END
10010 B=B+D:POKEA,D:A=A+1:NEXT:READD
10020 IFD<>BTHENPRINT"DATA ERROR IN LINE";PEEK(63)+256*PEEK(64):END:ELSELOOP
15000 INPUT"START ADDRESS";A:INPUT"END ADDRESS";F:INPUT"LINE NUMBER";D
15005 INPUT"STEP FOR LINE NUMBERS";H:INPUT"CHECK SUM (Y/N) ";A$
15010 PRINT"(CLR)";:FORC=1TO5:PRINTD"DATA";
15020 E=0:FORB=0TO15:PRINTPEEK(A+B);RIGHT$("(LEFT)(LEFT)(LEFT)(LEFT)(LEFT)",LEN(STR$(PEEK(A+B)))+1);
15021 IFBTHENPRINT",";:ELSEPRINT" ";
15022 PRINTRIGHT$("(RGHT)(RGHT)(RGHT)(RGHT)",LEN(STR$(PEEK(A+B)))-1);
15024 E=E+PEEK(A+B):NEXTB:D=D+H:A=A+16:IFA$="Y"THENPRINT","E"";
15025 IFA>FTHENPRINT",-1":GOTO15100
15030 PRINT:NEXTC:PRINT"A="A":D="D":F="F":H="H":A$="CHR$(34)A$CHR$(34)":GOTO15010"
15100 POKE239,C+1:POKE1319,19:FORG=1320TO1325:POKEG,13:NEXT
```

FROGGY



Daryl Bowers brings you close to the completion of your own arcade game.

THIS MONTH'S INSTALment brings us to the point where we have a playable game.

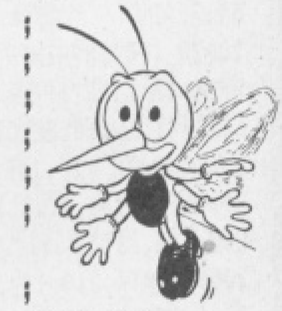


There are two more insertions into the main loop and a new "FINAL" routine.

'DEDCHK' simply calls the collision detection routines. The hardware sprite collision facility has a serious drawback — it tells you when one sprite has collided with another, but not which one it has collided with. This is fine for games such as Jet Set Willy, where any sprite collision indicates the death of the player, but in

```
1570                JSR DEAD
1580                JSR DEDCHK

11800 FINAL         JSR WAIT3
11810               JSR RVARS
11820               JSR INIT
11830               RTS
11840               ;
11850               ;
11860               ;
11870 DEDCHK
11880               ;
11890               JSR ROADCHK
11900               JSR BIKECHK
11910               JSR FLYCHK
11920               JSR BRDCHK
11930               ;
11940               ;
11950               RTS
11960               ;
11970               ;
11980               ;
11990 ROADCHK
12000               ;
12010               ;
12020               LDY STAGE
12030               LDA JUMPTYPE
12040               BNE BIG4
12050               ;
12060               LDA SPTAB,Y
12070               TAX
12080               LDA XTAB2,Y
12090               JMP TEST
12100               ;
12110 BIG4
12120               ;
12130               LDA SPTAB2,Y
12140               TAX
12150               LDA XTAB2B,Y
12160               ;
12170 TEST
12180               ;
12190               CPX #200
12200               BEQ CHKIT
12210               RTS
12220               ;
12230 CHKIT
12240               ;
12250               CLC
12260               ADC #48
12270               LSR A
12280               LSR A
12290               LSR A
12300               TAX
12310               LDY #7
12320 LOOP31
12330               LDA $06D0,X
12340               CMP #102
12350               BEQ GOTIT
12360               DEX
12370               DEY
12380               BNE LOOP31
12390 NOTIT         RTS
12400               ;
12410               ;
12420 GOTIT
12430               ;
12440               DEC RDDEDEL
12450               BNE NOTIT
12460               LDA #100
12470               STA RDDEDEL
12480               LDA FOOD
12490               CMP #'0'
12500               BEQ DEDED
12510               DEC FOOD
12520               RTS
12530 DEDED
12540               STA FOOD+1
12550               LDA #1
12560               STA DED
12570               RTS
12580               ;
12590               ;
12600               RTS
12610               ;
12620               ;
12630 BIKECHK
12640               ;
12650               ;
12660               ;
12670               ;
12680               ;
12690               LDA FRENCH
12700               CLC
12710               ADC #40
12720               STA $AC
12730               LDY STAGE
12740               LDX JUMPTYPE
12750               BNE BIG6
12760               LDA XTAB2,Y
12770               JMP TEST2
12780               ;
12790 BIG6
12800               LDA XTAB2B,Y
12810               ;
12820 TEST2
12830               ;
12840               CMP $AC
12850               BCC DEDED2
12860               RTS
12870               ;
12880               ;
12890 DEDED2
12900               LDA #1
12910               STA DED
12920               RTS
12930               ;
12940               ;
12950               ;
12960               ;
12970               ;
12980               ;
12990 FLYCHK
13000               ;
13010               LDY STAGE
13020               LDX JUMPTYPE
13030               BNE JUMPIN
13040 NOTFLY2       RTS
13050               ;
13060 JUMPIN
13070               ;
13080               LDA SPTAB2,Y
13090               CMP #204
13100               BNE NOTFLY2
13110               ;
13120               LDA FLYMOVE
13130               BEQ NOTFLY2
```



most games this will not suffice. For this reason I have detected collisions by comparing the X and Y co-ordinates of the Frog and the other characters.

'ROADCHK' is the first routine to be called. This checks if the Frog is sitting in a puddle. The code from the start to 'CHKIT' is used to ascertain that the Frog is sitting down. To check this the current sprite definition is found — if this is equal to 200 then the co-ordinates are checked.

To see whether the characters underneath him are puddle ones, we must first find the position of the relevant characters. Remember that the first visible sprite X co-ordinate is 31 and that the frog graphics start 17 pixels into the front sprite we must subtract 48 from the X position. If we divide this value by eight (the width of one character in pixels) then we have the X character position of the Frog.

'LOOP31' checks to see if the next seven characters are puddles or not, and if they are we go to 'GOTIT'. At this point we decrease the delay 'RDDEDEL' to reduce the damage sustained for each contact, then if this has reached zero we reduce the 'FOOD' left.

'BIKECHK' is a very simple check. We take the Frenchman's position, add 48, and store this in $AC. If the frog's X position is less than this, he is dead.

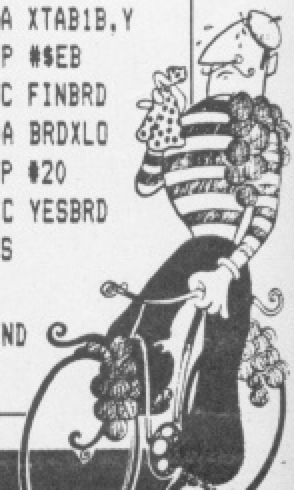
'FLYCHK' follows the same comparisons, with the addition that the Frog must be jumping, and if a collision takes place, the Fly's position is reset to the start again.

'SVARS' transfers the variable block to the end of the program in order that they can be retrieved at the start of a new game. 'RVARS' does exactly the opposite, and is called in 'FINAL' to reset all variables to their original values.

The final routine is 'BRDCHK' which operates in the same way as 'FLYCHK'.

Next month — frills.

```
13140                   ;
13150                   LDA FLYXHI
13160                   BNE HITEST
13170                   ;
13180                   LDA XTAB1B,Y
13190                   CLC
13200                   ADC #20
13210                   CMP FLYXLO
13220                   BCS NOTFLY2
13230                   CLC
13240                   ADC #15
13250                   BCS HITEST
13260                   CMP FLYXLO
13270                   BCC NOTFLY2
13280                   ;
13290 YESFLY            LDA #0
13300                   STA FLYMOVE
13310                   LDA #1
13320                   STA FLYXHI
13330                   LDA #255
13340                   STA FLYXLO
13350                   JSR PRNTFLY
13360                   LDA #'9'
13370                   STA FOOD
13380                   STA FOOD+1
13390                   LDX #2
13400                   JSR LOOP25
13410                   ;
13420 FINFLY            RTS
13430                   ;
13440                   ;
13450 HITEST
13460                   ;
13470                   LDA XTAB1B,Y
13480                   CMP #$EB
13490                   BCC FINFLY
13500                   LDA FLYXLO
13510                   CMP #20
13520                   BCC YESFLY
13530                   RTS
13540                   ;
13550                   ;
13560                   ;
13570                   ;
13580                   ;
13590 SVARS
13600                   ;
13610                   LDA #VARS&255
13620                   STA $FB
13630                   LDA #VARS/256
13640                   STA $FC
13650                   LDA #FINISH&255
13660                   STA $FD
13670                   LDA #FINISH/256
13680                   STA $FE
13690                   LDX #3
13700                   LDY #0
13710 LOOP36
13720                   LDA ($FB),Y
13730                   STA ($FD),Y
13740                   DEY
13750                   BNE LOOP36
13760                   INC $FC
13770                   INC $FE
13780                   DEX
13790                   BPL LOOP36
13800                   RTS
13810                   ;
13820 RVARS
13830                   ;
13840                   LDA #VARS&255
13850                   STA $FB
13860                   LDA #VARS/256
13870                   STA $FC
13880                   LDA #FINISH&255
13890                   STA $FD
13900                   LDA #FINISH/256
13910                   STA $FE
13920                   LDX #3
13930                   ;
13940                   LDY #0
13950 LOOP37
13960                   LDA ($FD),Y
13970                   STA ($FB),Y
13980                   DEY
13990                   BNE LOOP37
14000                   INC $FC
14010                   INC $FE
14020                   DEX
14030                   BEQ REMAIN
14040                   BPL LOOP37
14050                   RTS
14060 REMAIN            LDY #START&255
14070                   JMP LOOP37.
14080 BRDCHK
14090                   ;
14100                   LDY STAGE
14110                   LDX JUMPTYPE
14120                   BNE JUMPIN2
14130 NOTBRD2           RTS
14140                   ;
14150 JUMPIN2
14160                   ;
14170                   LDA SPTAB2,Y
14180                   CMP #204
14190                   BNE NOTBRD2
14200                   ;
14210                   LDA BRDMOVE
14220                   BEQ NOTBRD2
14230                   ;
14240                   LDA BRDXHI
14250                   BNE HITEST2
14260                   LDA XTAB1B,Y
14270                   CLC
14280                   ADC #20
14290                   CMP BRDXLO
14300                   BCS NOTBRD2
14310                   CLC
14320                   ADC #15
14330                   CMP BRDXLO
14340                   BCC NOTBRD2
14350                   ;
14360 YESBRD            LDA #1
14370                   STA DED
14380                   RTS
14390                   ;
14400 HITEST2
14410                   ;
14420                   LDA XTAB1B,Y
14430                   CMP #$EB
14440                   BCC FINBRD
14450                   LDA BRDXLO
14460                   CMP #20
14470                   BCC YESBRD
14480 FINBRD            RTS
14490 FINISH
14500                   .END
```



K Otton and A Adams add yet another dimension to your computer.



THE FOLLOWING PROGRAM is for use on files saved whilst using Telcom 64 (Your Commodore, April-May 1986). It provides an easy means of editing and changing a file with the choice to resave with a different, and perhaps, more appropriate name.

All the functions are available via the main menu (Figure 1) and perform the following tasks.

View File

This is a copy of the view file function found in Telcom 64. It allows the file to be viewed on the screen.

Use the space bar to pause the display and run-stop to exit back to the menu.

Edit File

This function gives access to the file (assuming one has been loaded in) and allows changes to be made.

For example, a file downloaded from a bulletin board could perhaps contain two pictures: (1) Snoopy, (2) pinup (see Figure 4).

Using the cursor left/right keys you can step through the file. To step through at high speed cursor up/down should be used as this steps in blocks of 255 bytes instead of single ones.

Delete and insert keys work the same as normal (although slower), however, for large deletes, F1 and F2 come into there own.

When the start of delete position is reached Press F1. "Delete from xxx" appears on the screen (where xxx = current position). Now step through to the end of delete position and Press F2. The portion between pressing F1 and F2 is now deleted and the file is viewed from the point where F1 was first pressed.

If an error was made when F1 was pressed it can be repressed to give a new start position .F2 selection is final and cannot be changed.

PROGRAM: EDITOR LOADER

```
10 IFA=0THENA=1:LOAD"EDITORC
000-CEE8",8,1
20 SYS49152
```

PROGRAM: EDITOR

```
2000 FORL=0TO238:CX=0:FORD=0
TO15:READA:CX=CX+A:POKE49152
+L*16+D,A:NEXTD
2010 READA:IFA<>CXTHENPRINT"
ERROR IN LINE";2040+(L*10):S
TOP
2020 NEXTL
2040 DATA76,207,201,0,157,0,
48,232,208,250,32,68,229,32,
15,201,1956
2050 DATA169,8,32,210,255,16
9,14,32,210,255,162,0,189,11
6,192,157,2170
2060 DATA130,4,189,135,192,1
57,170,4,189,154,192,157,250
,4,189,173,2289
2070 DATA192,157,74,5,189,19
2,192,157,154,5,189,211,192,
157,234,5,2305
2080 DATA189,230,192,157,58,
6,189,249,192,157,138,6,189,
12,193,157,2314
2090 DATA218,6,189,31,193,15
7,42,7,232,224,19,208,191,96
,162,0,1975
2100 DATA169,1,157,0,216,157
,0,217,157,0,218,157,0,219,2
32,224,2124
2110 DATA0,208,239,96,42,42,
32,32,69,68,73,84,79,82,32,7
7,1255
2120 DATA69,78,85,32,32,42,4
2,46,46,46,46,46,46,46,46
,794
2130 DATA46,46,46,46,46,46,4
6,46,46,46,49,32,86,73,69,87
,856
2140 DATA32,70,73,76,69,46,4
6,46,46,46,46,70,49,50,32,69
,866
2150 DATA68,73,84,32,70,73,7
6,69,46,46,46,46,46,46,70,50
,941
2160 DATA51,32,70,73,76,69,3
2,76,69,78,71,84,72,46,46,46
,991
2170 DATA46,70,51,52,32,76,7
9,65,68,32,70,73,76,69,46,46
,951
2180 DATA46,46,46,46,70,52,5
3,32,83,65,86,69,32,70,73,76
,945
2190 DATA69,46,46,46,46,46,4
6,70,53,54,32,72,69,88,32,84
,899
2200 DATA79,32,66,65,83,73,6
7,46,46,46,70,54,55,32,80,82
,976
2210 DATA73,78,84,32,70,73,7
6,69,46,46,46,46,46,70,55,56
,966
2220 DATA32,69,88,73,84,32,8
4,79,32,66,65,83,73,67,46,46
,1019
2230 DATA70,56,255,32,68,229
,173,0,48,208,20,160,193,169
,205,32,1918
2240 DATA30,171,160,193,169,
185,32,30,171,32,228,255,240
,251,96,169,2412
2250 DATA0,133,251,169,48,13
3,252,32,94,192,160,0,177,25
1,208,3,2103
2260 DATA76,165,193,32,210,2
55,32,133,193,32,117,193,230
,251,208,2,2322
2270 DATA230,252,76,92,193,1
69,20,141,234,193,160,255,13
6,208,253,206,2818
2280 DATA234,193,208,246,96,
```

```
32,228,255,240,11,201,3,208,
3,76,162,2396
2290 DATA193,201,32,240,1,96
,32,228,255,201,32,208,1,96,
201,3,2020
2300 DATA208,244,104,104,96,
160,193,169,218,32,30,171,16
0,193,169,185,2436
2310 DATA32,30,171,32,228,25
5,240,251,96,13,17,17,17,32,
80,82,1593
2320 DATA69,83,83,32,65,78,8
9,32,75,69,89,32,34,17,17,32
,896
2330 DATA32,78,79,32,70,73,7
6,69,32,34,13,13,32,69,78,68
,848
2340 DATA32,79,70,32,70,73,7
6,69,32,34,0,0,0,170,4,190,9
31
2350 DATA85,255,169,0,133,25
1,169,48,133,252,160,0,177,2
51,240,9,2332
2360 DATA230,251,208,2,230,2
52,76,252,193,165,251,141,19
2,194,165,252,3054
2370 DATA141,193,194,96,32,2
42,193,32,68,229,160,194,169
,196,32,30,2201
2380 DATA171,162,0,169,48,32
,205,189,32,145,194,169,48,3
2,159,194,1949
2390 DATA169,0,32,159,194,16
0,194,169,216,32,30,171,174,
192,194,173,2259
2400 DATA193,194,32,205,189,
32,145,194,173,193,194,32,15
9,194,173,192,2494
2410 DATA194,32,159,194,160,
195,169,6,32,30,171,56,169,3
,237,192,1999
2420 DATA194,141,194,194,169
,159,237,193,194,141,195,194
,174,194,194,173,2940
2430 DATA195,194,32,205,189,
32,145,194,173,195,194,32,15
9,194,173,194,2500
2440 DATA194,32,159,194,160,
193,169,185,32,30,171,32,228
,255,240,251,2525
2450 DATA96,169,32,32,210,25
5,32,210,255,169,36,32,210,2
55,96,72,2161
2460 DATA74,74,74,74,201,10,
144,2,105,6,105,48,32,210,25
5,104,1518
2470 DATA41,15,201,10,144,2,
105,6,105,48,32,210,255,96,0
,0,1270
2480 DATA0,48,3,111,5,29,17,
17,83,84,65,82,84,32,65,68,7
93
2490 DATA68,82,69,83,83,32,3
2,34,13,29,17,17,69,78,68,32
,806
2500 DATA65,68,68,82,69,83,8
3,32,32,32,32,34,13,17,17,32
,759
2510 DATA84,79,84,65,76,32,6
6,89,84,69,83,32,32,32,32,34
,973
2520 DATA19,34,19,36,36,34,1
3,17,17,32,73,78,83,69,82,84
,726
2530 DATA83,32,76,69,70,84,3
2,32,32,34,32,34,13,17,17,32
,689
2540 DATA84,79,84,65,76,32,6
6,89,84,69,83,32,32,32,32,34
,973
2550 DATA19,34,19,36,36,34,1
3,17,17,32,73,78,83,69,82,84
,726
2560 DATA83,32,76,69,70,84,3
2,32,32,34,170,169,48,133,25
2,169,1485
2570 DATA4,133,251,169,8,133
,254,169,1,133,253,160,0,177
,251,240,2336
2580 DATA46,201,13,240,64,14
1,246,195,200,177,251,141,24
5,195,32,174,2561
2590 DATA195,173,247,195,160
,0,145,253,24,165,251,105,2,
133,251,165,2464
2600 DATA252,105,0,133,252,2
4,230,253,208,2,230,254,76,9
1,195,24,2329
2610 DATA165,253,105,2,133,4
5,165,254,105,0,133,46,169,0
,200,145,1920
2620 DATA253,200,145,253,96,
230,251,208,2,230,252,76,91,
195,169,0,2651
2630 DATA141,247,195,173,246
,195,201,58,144,15,41,15,24,
105,9,10,1819
2640 DATA10,10,10,141,243,19
5,76,210,195,41,15,10,10,10,
10,141,1327
2650 DATA243,195,173,245,195
,201,58,144,11,41,15,24,105,
9,141,244,2044
2660 DATA195,76,233,195,41,1
5,141,244,195,173,244,195,13
,243,195,141,2539
2670 DATA247,195,96,96,14,85
,93,110,85,255,85,95,16,255,
85,32,1844
2680 DATA231,255,173,149,197
,162,1,201,84,240,2,162,8,16
0,0,169,2194
2690 DATA1,32,186,255,169,17
8,160,197,32,30,171,162,15,1
42,157,197,2084
2700 DATA32,243,196,192,0,20
8,7,173,149,197,201,84,208,2
42,173,149,2454
2710 DATA197,201,68,208,66,1
69,64,141,20,2,169,48,141,21
,2,169,1686
2720 DATA58,141,22,2,160,0,1
85,0,2,153,23,2,200,204,155,
197,1504
2730 DATA208,244,169,44,153,
23,2,169,80,153,24,2,173,148
,197,201,1990
2740 DATA83,208,12,169,44,15
3,25,2,169,87,153,26,2,200,2
00,200,1733
2750 DATA200,200,200,200,76,
138,196,172,155,197,240,14,1
60,0,185,0,2333
2760 DATA2,153,20,2,200,204,
155,197,208,244,152,162,20,1
60,2,32,1913
2770 DATA189,255,169,160,133
,178,96,169,76,141,148,197,1
69,191,141,0,2412
2780 DATA2,32,255,195,169,0,
162,0,160,48,32,213,255,176,
41,96,1836
2790 DATA169,83,141,148,197,
169,191,141,0,2,32,255,195,1
69,0,133,2025
2800 DATA251,169,48,133,252,
174,151,197,172,150,197,169,
251,32,216,255,2817
2810 DATA176,6,32,183,255,20
8,1,96,32,231,255,169,125,16
0,197,32,2158
2820 DATA30,171,169,185,160,
193,32,30,171,32,237,196,96,
32,228,255,2217
2830 DATA240,251,96,142,154,
197,232,169,46,32,210,255,20
2,224,0,208,2658
2840 DATA248,174,154,197,232
,169,157,32,210,255,202,224,
0,208,248,169,2879
2850 DATA32,162,0,157,1,2,23
2,224,20,208,248,160,0,140,1
53,197,1936
2860 DATA169,164,32,210,255,
169,157,32,210,255,32,228,25
5,240,251,172,2831
2870 DATA153,197,141,152,197
,169,46,32,210,255,169,157,3
2,210,255,173,2548
2880 DATA152,197,201,13,240,
45,201,20,208,13,192,0,240,2
07,136,169,2234
```

Run-stop will place an end of file marker (chequered flag) at the position above the arrow, this point cannot be passed until it is over written with a space or charactor.

Shifted run-stop will exit to the menu. Pressing any valid key will overwrite the original character above the arrow and then step on to the next.

Remember in this section an up arrow (↑) denotes a return and can affect both the printout and viewfile displays.

File Length

Displays the current file's start/end address in decimal/hex and the number of spare bytes available should you wish to type in more details.

Load File

Fairly obvious this one, but it should be noted that only one error message is displayed for both disk and tape regardless of the actual error. It is "ERROR LOAD/SAVE" and could be any fault from 'file not found' on disk to the tape stop key being pressed on tape player.

The filename must be known, although the file symbol will be placed in front automatically as it is in Telcom 64.

Save File

This will resave the current file up to the chequered end of file marker. Any alterations made in edit mode (option two) will be saved and the length of file can be made longer or shorter by moving the end of file market. This allows one log file to be split into shorter ones with only the required information left in them.

Hex to Basic

If a hex file has been downloaded (Figure 2) it should be edited so that the first digit is at current position zero. An end of file marker should be placed at the end of the hex.

By selecting option six, the hex will be converted to ASCII and placed at 2048 onwards to build up a Basic program in memory. On

completion the program exits to Basic and the new program can be listed or saved as normal.

Print File

This is an option to print out the new edited file for a hard copy. Again no check is made to see if the device is present. A choice of printer is given when the program is first started up.

Exit to Basic

This one speaks for itself. Restart FILE EDITOR with SYS49152.

Loader

For people using an 801/803 printer, only lines 10 and 50 are needed. 1520 users should enter the other lines depending on their requirements. The option of device four for 1520 is only for those with the modified 1520.

```
10 IF A = 0 THEN A = 1 : LOAD
"FILE EDITOR",8,1
20 X = ?:REM 6 OR 4 FOR
DEVICE NUMBER
30 OPEN 6,X,6:PRINT#
6,1:CLOSE6:REM SET LOWER
CASE/SHIFTED UPPER CASE
40 OPEN 3,X,3:PRINT#
3,4:CLOSE3: REM SET To 80
COL MODE
50 SYS 49152
```

Figure 1

Main Menu

1) VIEW FILE F1
2) EDIT FILE F2
3) FILE LENGTH F3
4) LOAD FILE F4
5) SAVE FILE F5
6) HEX TO BASIC F6
7) PRINT FILE F7
8) EXIT TO BASIC F8

Figure 2

Hex Example

FROM... Type P to Pause, S to Stop. ↑↑↑↑0801000A2A2 A2A2 A2A2A2A2A2A2A2A--
EDIT TO. 0801000A2A2A2 A2A2A2A2A2A2A----

Note
Select option 6 and then save or list as normal.

```
2890 DATA157,32,210,255,76,2
9,197,41,127,201,32,144,192,
204,157,197,2251
2900 DATA240,187,173,152,197
,153,1,2,32,210,255,169,0,13
3,212,200,2316
2910 DATA76,29,197,200,140,1
55,197,169,13,32,210,255,96,
147,32,17,1965
2920 DATA69,82,82,79,82,32,7
9,78,32,83,65,86,69,47,76,79
,1120
2930 DATA68,68,58,34,76,68,0
,0,13,3,15,4,8,15,64,48,542
2940 DATA58,83,48,48,32,32,3
2,32,32,32,32,32,32,32,32,32
,621
2950 DATA32,34,147,5,70,73,7
6,69,32,78,65,77,69,32,34,32
,925
2960 DATA32,32,32,0,0,170,0,
170,0,170,19,17,17,17,17,17,
710
2970 DATA17,17,17,17,29,29,2
9,29,29,29,29,29,29,29,83,69
,510
2980 DATA76,69,67,84,32,84,6
5,80,69,32,79,82,32,68,73,83
,1075
2990 DATA75,32,63,13,13,13,2
9,32,32,32,32,32,32,32,32,32
,526
3000 DATA32,32,32,32,69,78,8
4,69,82,32,84,32,79,82,32,68
,919
3010 DATA34,19,17,17,17,29,3
2,32,32,32,32,32,32,32,32,32
,453
3020 DATA32,32,83,69,76,69,6
7,84,32,80,82,73,78,84,69,82
,1092
3030 DATA13,29,32,32,32,32,3
2,32,32,32,32,32,32,32,45,45
,516
3040 DATA45,45,45,45,45,45,4
5,45,45,45,45,45,13,13,29,32
,627
3050 DATA32,32,49,46,46,56,4
8,49,47,56,48,51,32,32,32,32
,688
3060 DATA40,49,53,50,53,41,1
3,13,29,32,32,32,50,46,46,49
,628
3070 DATA53,50,48,32,68,69,8
6,32,54,32,40,80,82,73,78,84
,961
3080 DATA69,82,32,80,76,79,8
4,84,69,82,41,13,13,29,32,32
,897
3090 DATA32,51,46,46,49,53,5
0,48,32,68,69,86,32,52,32,40
,786
3100 DATA45,45,45,45,45,68,7
3,84,84,79,45,45,45,45,45,41
,879
3110 DATA13,13,13,13,29,32,3
2,32,32,32,32,32,32,32,32,69
,470
3120 DATA78,84,69,82,32,49,4
4,50,32,79,82,32,51,46,34,20
8,1052
3130 DATA32,30,32,61,32,82,6
9,84,85,82,78,32,32,32,32,32
,827
3140 DATA32,32,102,32,61,
32,69,78,68,32,79,70,32,70,7
3,894
3150 DATA76,69,32,32,102,255
,255,255,255,255,255,255,255
,255,255,255,3116
3160 DATA255,30,255,255,255,
255,255,255,255,255,255,255,
255,255,255,255,3855
3170 DATA255,255,255,255,32,
33,34,35,36,37,38,39,40,41,4
2,43,1470
3180 DATA44,45,46,47,48,49,5
0,51,52,53,54,55,56,57,58,59
,824
3190 DATA60,61,62,63,0,1,2,3
,4,5,6,7,8,9,10,11,312
3200 DATA12,13,14,15,16,17,1
8,19,20,21,22,23,24,25,26,27
,312
3210 DATA28,29,30,31,96,65,6
6,67,68,69,70,71,72,73,74,75
,984
3220 DATA76,77,78,79,80,81,8
2,83,84,85,86,87,88,89,90,12
3,1368
3230 DATA124,125,126,127,128
,129,130,131,132,133,134,135
,136,137,138,139,2104
3240 DATA140,141,142,143,144
,145,146,147,148,149,150,151
,152,153,154,155,2360
3250 DATA156,157,158,159,160
,161,162,163,164,165,166,167
,168,169,170,171,2616
3260 DATA172,173,174,175,176
,177,178,179,180,181,182,183
,184,185,186,187,2872
3270 DATA188,189,190,191,192
,65,66,67,68,69,70,71,72,73,
74,75,1720
3280 DATA76,77,78,79,80,81,8
2,83,84,85,86,87,88,89,90,21
9,1464
3290 DATA220,221,222,223,224
,225,226,227,228,229,230,231
,232,233,234,235,3640
3300 DATA236,237,238,239,240
,241,242,243,244,245,246,247
,248,249,250,251,3896
3310 DATA252,253,254,255,255
,255,255,255,255,255,255,255
,255,255,255,255,4074
3320 DATA255,13,255,255,255,
255,255,255,255,255,255,255,
255,255,255,255,3838
3330 DATA255,255,255,255,32,
33,34,35,36,37,38,39,40,41,4
2,43,1470
3340 DATA44,45,46,47,48,49,5
0,51,52,53,54,55,56,57,58,59
,824
3350 DATA60,61,62,63,64,65,6
6,67,68,69,70,71,72,73,74,75
,1080
3360 DATA76,77,78,79,80,81,8
2,83,84,85,86,87,88,89,90,91
,1336
3370 DATA92,93,94,95,0,65,66
,67,68,69,70,71,72,73,74,75,
1144
3380 DATA76,77,78,79,80,81,8
2,83,84,85,86,87,88,89,90,91
,1336
3390 DATA92,93,94,95,255,0,0
,255,255,255,255,255,255,255
,255,255,2924
3400 DATA255,255,255,255,255
,255,255,255,255,255,255,255
,255,255,255,255,4080
3410 DATA255,255,255,255,32,
255,255,255,255,255,255,255,
255,255,255,255,3857
3420 DATA255,255,255,255,255
,255,255,255,255,255,255,255
,255,255,255,255,4080
3430 DATA255,255,255,255,255
,193,194,195,196,197,198,199
,200,201,202,203,3453
3440 DATA204,205,206,207,208
,209,210,211,212,213,214,215
,216,217,218,255,3420
3450 DATA255,255,255,255,255
,255,255,255,255,255,255,255
,255,255,255,255,4080
3460 DATA255,255,255,255,255
,255,255,255,255,255,255,255
,255,255,255,255,4080
3470 DATA255,255,255,255,255
,255,255,255,255,255,255,255
,255,255,255,255,4080
3480 DATA255,255,255,255,240
,163,201,131,208,1,96,201,15
7,208,26,32,2684
3490 DATA94,192,162,0,169,42
,157,0,4,157,192,7,232,224,4
```

```
0,208,1880
3500 DATA245,169,40,133,251,
169,4,133,252,169,42,160,0,1
45,251,160,2323
3510 DATA39,145,251,165,251,
24,105,40,133,251,165,252,10
5,0,133,252,2311
3520 DATA165,252,201,7,208,2
27,165,251,201,192,208,221,9
6,201,201,32,2828
3530 DATA173,195,201,201,49,
208,3,76,100,201,201,50,208,
3,76,109,2054
3540 DATA201,76,118,201,169,
4,162,4,160,7,76,127,201,169
,4,162,1841
3550 DATA6,160,0,76,127,201,
169,4,162,4,160,0,76,127,201
,32,1505
3560 DATA186,255,169,0,32,18
9,255,32,192,255,169,0,133,1
06,169,48,2190
3570 DATA133,107,160,0,177,1
06,240,23,141,194,201,162,4,
32,201,255,2136
3580 DATA173,194,201,32,210,
255,230,106,208,232,230,107,
76,146,201,162,2763
3590 DATA4,32,201,255,169,13
,32,210,255,32,204,255,169,4
,32,195,2062
3600 DATA255,96,71,51,76,93,
201,32,68,229,96,32,151,196,
96,32,1775
3610 DATA68,229,169,1,141,13
4,2,32,129,203,32,47,203,32,
10,192,1624
3620 DATA32,228,255,240,251,
201,133,208,9,32,51,193,32,1
0,192,76,2143
3630 DATA224,201,201,137,208
,9,32,124,202,32,10,192,76,2
24,201,201,2274
3640 DATA134,208,9,32,20,194
,32,10,192,76,224,201,201,13
8,208,9,1888
3650 DATA32,78,202,32,10,192
,76,224,201,201,135,208,9,32
,82,202,1916
3660 DATA32,10,192,76,224,20
1,201,139,208,9,32,111,202,3
2,10,192,1871
3670 DATA76,224,201,201,136,
208,9,32,120,202,32,10,192,7
6,224,201,2144
3680 DATA201,140,208,3,76,74
,202,76,224,201,32,68,229,96
,32,151,2013
3690 DATA196,96,32,140,203,1
69,0,160,1,145,253,230,253,2
08,2,230,2318
3700 DATA254,165,253,141,151
,197,165,254,141,150,197,32,
176,196,96,32,2600
3710 DATA75,195,32,68,229,10
4,104,96,32,80,201,96,32,68,
229,162,1803
3720 DATA40,169,1,157,223,21
7,169,96,157,223,5,202,208,2
43,169,30,2309
3730 DATA141,28,6,169,7,141,
28,218,160,0,169,32,153,236,
47,200,1735
3740 DATA192,20,208,248,32,2
23,202,169,47,133,252,169,23
6,133,251,173,2688
3750 DATA0,48,240,10,32,191,
203,32,140,203,32,208,203,96
,162,0,1800
3760 DATA189,31,206,157,244,
5,232,224,9,208,245,160,206,
169,40,32,2357
3770 DATA30,171,160,193,169,
190,32,30,171,32,228,255,240
,251,96,160,2408
3780 DATA0,169,1,153,43,219,
153,123,219,185,208,198,153,
43,7,185,2059
3790 DATA226,198,153,123,7,2
00,192,18,208,231,96,165,251
,141,27,206,2442
3800 DATA165,252,141,26,206,
173,27,206,24,105,20,170,173
,26,206,105,2025
3810 DATA0,141,26,206,56,233
,48,141,26,206,32,205,189,16
9,32,32,1742
3820 DATA210,255,32,210,255,
32,210,255,32,210,255,32,210
,255,96,32,2581
3830 DATA15,201,169,202,160,
197,32,30,171,32,228,255,240
,251,201,84,2468
3840 DATA240,4,201,68,208,24
3,141,149,197,32,68,229,32,1
5,201,169,2197
3850 DATA17,160,198,32,30,17
1,32,228,255,240,251,201,49,
208,3,76,2151
3860 DATA109,203,201,50,208,
3,76,109,203,201,51,208,233,
141,195,201,2392
3870 DATA96,165,251,24,105,2
0,141,25,206,165,252,105,0,1
41,24,206,1926
3880 DATA96,162,0,169,0,157,
0,48,232,208,250,96,169,48,1
33,254,2022
3890 DATA169,0,133,253,160,0
,177,253,240,9,230,253,208,2
,230,254,2571
3900 DATA76,150,203,165,253,
141,17,206,165,254,141,16,20
6,56,173,17,2239
3910 DATA206,233,20,141,19,2
06,173,16,206,233,0,141,18,2
06,96,160,2074
3920 DATA0,177,251,170,189,2
44,198,153,224,5,200,192,40,
208,242,96,2589
3930 DATA169,144,160,206,32,
30,171,32,251,202,169,170,16
0,206,32,30,2164
3940 DATA171,174,17,206,173,
16,206,56,233,48,32,205,189,
169,32,32,1959
3950 DATA210,255,32,210,255,
32,210,255,32,210,255,169,19
4,160,206,32,2717
3960 DATA30,171,169,3,56,237
,17,206,170,169,159,237,16,2
06,142,29,2017
3970 DATA206,141,28,206,32,2
05,189,169,32,32,210,255,32,
210,255,32,2234
3980 DATA210,255,32,210,255,
32,210,255,32,228,255,240,16
3,201,131,208,2917
3990 DATA1,96,201,157,208,26
,165,252,201,47,208,6,165,25
1,201,236,2421
4000 DATA240,11,165,251,208,
2,198,252,198,251,32,191,203
,76,208,203,2689
4010 DATA201,29,208,38,165,2
52,201,159,208,6,165,251,201
,3,240,23,2350
4020 DATA165,252,205,18,206,
208,7,165,251,205,19,206,240
,9,230,251,2637
4030 DATA208,2,230,252,32,19
1,203,76,208,203,201,148,208
,82,173,29,2446
4040 DATA206,208,5,173,28,20
6,240,69,165,251,24,105,19,1
41,15,206,2061
4050 DATA165,252,105,0,141,1
4,206,169,159,133,254,169,3,
133,253,160,2316
4060 DATA0,177,253,160,1,145
,253,165,253,208,2,198,254,1
98,253,165,2685
4070 DATA254,205,14,206,208,
233,165,253,205,15,206,208,2
26,169,32,160,2759
4080 DATA20,145,251,32,191,2
03,32,140,203,169,0,133,198,
76,208,203,2204
4090 DATA201,133,208,28,165,
251,141,20,206,165,252,141,2
1,206,169,75,2382
4100 DATA160,206,32,30,171,1
69,255,141,30,206,32,251,202
,76,208,203,2372
4110 DATA201,137,208,107,173
,30,206,240,99,165,251,133,8
7,165,252,133,2587
4120 DATA88,173,20,206,133,8
9,173,21,206,133,90,56,165,8
8,229,90,1960
4130 DATA240,5,176,12,76,92,
```

At first the edit function (F2) view of a file may seem strange. Different boards send out an assortment of codes to make up a screen.

Figure 3

Note: *= return
---15-spaces---WELCOME TO---15-spaces---
---15-spaces---TELCOM 64*

Figure 3A
---15-spaces---WELCOME TO*
---15-spaces---TELCOM 64*

Note

The screen layouts shown in Figure 3 and 3A will both give the same result on screen. However, when viewed, the difference is quite considerable. In Figure 3 there are 15 spaces each side of 'WELCOME TO' then another 15 to 'TELCOM 64' giving a total of 65 bytes. In Fig 3A the file only has 51 bytes as a return has been placed after the 'TO' which causes the cursor to return to the start of the next line.

As already mentioned this will differ between different boards and viewing the file is the only way to know which method is being used.

Figure 3A

Type Ctrl X to Abort Download ↑↑↑↑ ----snoopy data---- ↑↑↑↑
↑ current position 0: ↑ current position 29
Press F1 at current pos 0 and F2 at current pos 29 the file will now look like-
↑↑↑↑----snoopy data----↑↑↑↑

Note

By placing an end of file market () at the point shown and then saving the file our new file will only contain Snoopy.

This procedure can be carried out as many times as required.

```
205,56,165,87,229,89,240,62,144,60,1938
4140 DATA160,20,177,87,160,20,145,89,230,87,208,2,230,88,230,89,2022
4150 DATA208,2,230,90,165,90,201,159,208,230,165,89,201,3,208,224,2473
4160 DATA173,20,206,133,251,173,21,206,133,252,169,0,141,30,206,169,2283
4170 DATA97,160,206,32,30,171,32,191,203,32,140,203,76,208,203,201,2185
4180 DATA20,208,50,32,140,203,165,251,133,253,165,252,133,254,160,21,2440
4190 DATA177,253,160,20,145,253,230,253,208,2,230,254,165,254,201,159,2964
4200 DATA208,236,165,253,201,3,208,230,32,191,203,32,140,203,169,0,2474
4210 DATA133,198,76,208,203,201,3,208,15,169,0,160,20,145,251,32,2022
4220 DATA140,203,32,191,203,76,208,203,201,17,208,18,32,113,203,173,2221
4230 DATA24,206,201,48,240,2,198,252,32,191,203,76,208,203,201,145,2430
4240 DATA208,35,32,113,203,238,24,206,173,24,206,205,16,206,240,5,2134
4250 DATA176,13,76,221,205,173,25,206,205,17,206,176,2,230,252,32,2215
4260 DATA191,203,76,208,203,72,165,252,201,159,208,6,165,251,201,3,2564
4270 DATA240,25,104,170,189,244,199,201,255,240,16,160,20,145,251,230,2689
4280 DATA251,208,2,230,252,32,191,203,32,140,203,76,208,203,0,0,2231
4290 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,32,32
4300 DATA78,79,32,70,73,76,69,32,19,17,17,17,29,29,29,29,695
4310 DATA29,29,29,29,29,29,32,32,32,32,32,32,32,32,32,32,494
4320 DATA157,157,157,157,157,157,157,157,157,157,34,19,17,17,17,17,1691
4330 DATA17,17,17,32,100,101,108,101,116,101,32,102,114,111,109,32,1210
4340 DATA34,19,17,17,17,17,17,17,17,32,32,32,32,32,32,32,396
4350 DATA32,32,32,32,32,32,32,32,32,32,32,32,34,19,17,17,471
4360 DATA17,29,29,29,32,32,32,32,32,32,32,32,32,32,34,0,458
4370 DATA5,19,17,17,17,32,99,117,114,114,101,110,116,32,112,111,1133
4380 DATA115,105,116,105,111,110,46,46,46,34,5,19,17,32,102,105,1114
4390 DATA108,101,32,101,110,100,32,97,100,100,114,101,115,115,46,46,1418
4400 DATA46,34,5,19,17,17,17,17,17,32,110,117,109,32,105,110,804
4410 DATA115,101,114,116,115,32,108,101,102,116,46,46,46,34,234,234,1660
4420 DATA234,234,234,234,234,234,234,255,186,27,160,0,108,111,28,1,2514
5000 REM ******************************
5010 REM *   READY TO SAVE CODE       *
5020 REM ******************************
5030 REM CHANGE ,8 TO ,1 IN LINE BELOW IF USING CASSETTE.
5040 POKE43,0:POKE44,192:POKE45,235:POKE46,206:SAVE"EDITORC000-CEE8",8
```

REMEMBER the PLUS/4

Joe Bradley delves into the Plus/4's memory and looks at machine code programs.

MANY NEWCOMERS TO computing may have bought a Plus/4 and have been disappointed that very few articles have been written for this machine. This article is for those new enthusiasts who wish to look into the machine and start writing machine code routines.

There is some difficulty in obtaining a full memory map for the Plus/4 but this article is intended to help in making a start.

First let us have a look at the different sections of the memory – owners of a Plus/4 are fortunate here because the computer contains an inbuilt monitor which will help.

If you wish to write machine code routines you must become familiar with the way the monitor works and the different commands that are available.

The computer contains two different types of memory location, those that you can change, called Random Access Memory or RAM, and memory locations that are Read Only Memory or ROM, these are used by the operating system and cannot be altered.

Switch on your Plus/4 and type:

MONITOR [RETURN]

the computer will respond with

MONITOR

PC SR AC XR YR SP
; FFFF 00 FF FF FF F9

or something similar.

The abbreviations are:

PC Program Counter	This is the current address being processed by the computer.
SR Status Register	This contains six flags which give information about the current status of the processor.
AC Accumulator the work horse of machine code rout routines	
XR X Register	
YR Y Register	
SP Stack Pointer	This gives the next free location on the stack which is a temporary storage area used during processing.

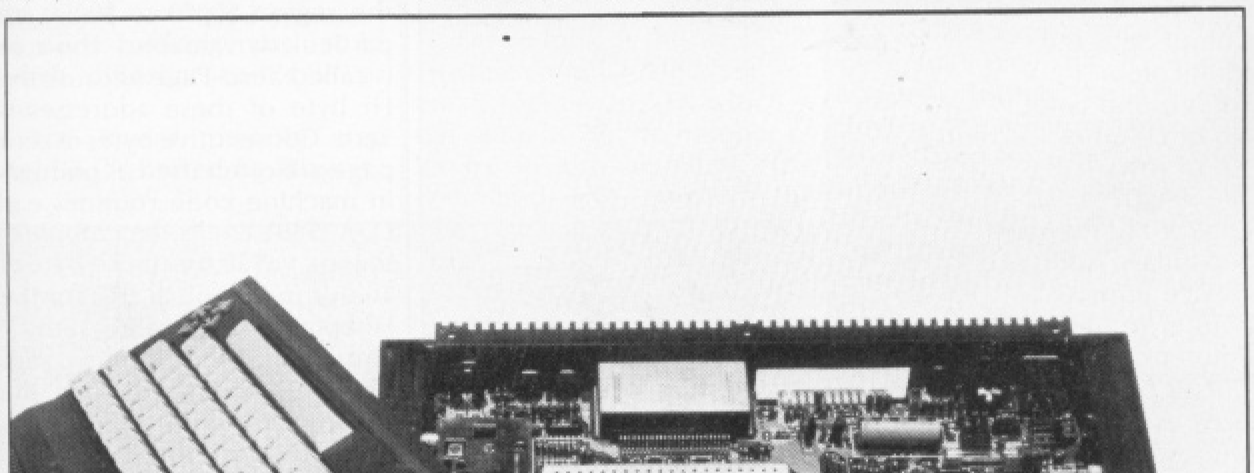
Now type:

M0000 003F [RETURN]

If a question mark is printed you have made an error in entry, probably you have typed letter O instead of a 0 (Zero number).

If the entry was correct you will see displayed eight rows of numbers these are the numbers stored in the memory locations 0000 to 003F. Note addresses are usually given in Hexadecimal code which counts in, units, 16s, 256s and 4096s. Thus 003F is 3 X 16 and another is 15 i.e. 63 in normal decimal numbers.

Each location can hold a number from zero to 255 this is called a byte and is made up of eight binary 'bits' which can be



either zero or one. More details of this may be read in any book of machine code.

The block on the extreme right in reverse print is the ASCII dump of the code if it is possible to be printed. When a character cannot be printed it is displayed as a full stop (.).

The highest address that can be read by the processor is $FFFF. The dollar sign shows that this is a hexadecimal number and is equal to (15 X 4096) + (15 X 256) + (15 X 16) + 15. In decimal this works out as 64 X 1024 which is almost 64,000 and is the reason most early computers were limited to 64K (approximately).

All the locations from 0000 to 003F which are on the screen are RAM locations that you can change but only with care! The first 4K (actually 4 X 1024) bytes of memory are used by the operating system and problems can arise if you can change the value. Let's examine this in more detail.

Locations 002B and 002C (43 and 44 decimal) are pointers to the start of Basic. To find their values, read down the left hand side numbers to 0028 and then move right counting 8, 9, A, B, C. The $2B and $2C locations should contain the values of 01 10. If you have not written any machine code then this is your first surprise in that addresses are always stored in the computer in what appears to be the wrong order, what is called the Lo (Low) byte first and then the Hi (High) byte. The pointers tell us that the current start to a Basic program is $1001.

Now let us try to alter these values. Move the cursor over the 01 at location $3B and change it to 06. Press [RETURN]. The value will now be changed and when we return to Basic the computer will take $1006 to be the start of Basic.

Return to Basic by typing a letter X and then [RETURN]. The computer prints READY and everything looks OK. However, enter a Basic line, say:

1 PRINT [RETURN]

Your screen will go haywire and nothing you do with the keyboard will bring back control. We say the computer has hung. Now we see another advantage of the Plus/4 over many other computers, it has a RESET key. This key is at the side next to the ON/OFF switch. Press this small button and the computer will be reset without switching off. Later when you are developing machine code routines your computer may sometimes hang but you will be able to keep your programs intact if you hold down the RUN/STOP key while you press the RESET button. (A very valuable feature which you will learn to treasure).

Your computer is now reset and you now know that some parts of RAM cannot be altered without thought! As you develop your machine code expertise you will need to know which locations you can use without dire consequences. The abbreviated table of memory locations from 0000 to $0FFF shows some that I have found useful, a safe rule is to restore the original value after use.

Consecutive locations in the region $0000 to $00FF are particularly valuable – this area is called Zero-Page because the Hi byte of these addresses is zero. Consecutive bytes in zero page are often used as pointers in machine code routines e.g. LDA ($3B),Y tells the computer to look at $3B for the Lo byte of an address, look at $3C for the Hi byte, add the value of the Y register to the address obtained and then load the accumulator with what it finds at the calculated address.

We have seen that Basic normally starts at $1001 but if you type

GRAPHIC1 [RETURN]

The screen will show a haphazard pattern because you will have moved into the High Resolution mode. This mode takes an extra 12K of RAM memory and the computer gets this by moving the start of Basic up to $4001. Even though you may not be able to see on your screen

System Memory Locations

ADDRESS HEX	DECIMAL	SYSTEM USE	AVAILABLE FOR USE
$0000-$0001	0-1	Input output chip	NO, Leave alone
$002-$0006	2-7	Temp. Used in search and renumber routines	YES
$0008	8	Flag used in quote scan	YES
$0014-80015	20-21	Temp-integer evaluation	YES
$002B-$002C	43-44	Start of Basic	YES - but reset
$002d-$002E	45-46	Start of Basic Variables	YES - but reset
$002F-$0030	47-48	Start of Basic Arrays	YES
$0031-$0032	49-50	End of Basic Arrays	YES
$0033-$0034	51-52	Bottom of strings	YES
$0035-$0036	53-54	String pointer	YES
$0037-$0038	55-56	Top of available memory	Only move down to protect memory
$0039-$003A	57-58	Current line number	YES
$003B-$003C	59-60	Pointers used in get character routine	YES
$003D-$0042	61-66	Pointers in ROM routines	YES
$0083	131	Graphic mode	NO
$0084	132	Colour Selected	
$0085	133	Multicolour 1	
$0086	134	Foreground colour	
$0087	135	No of columns - screen	
$0088	136	No of rows	
$00AB	171	File length	
$00AC	172	Logical file number	
$00AD	173	Secondary Address	
$00AE	174	Device number	
$00AF-$00B0	175-176	Pointer to file name	
$00C8-$00C9	200-201	Pointer current screen line	
$00CA	202	Cursor column	
$0100-$010F	256-271	Area used to store string after number conversion	
$0124-$01FF	291-511	STACK	
$0333-$03F2	819-1010	Cassette tape buffer	Very useful to store short machine code.
$0509-$0512	1289-1298	Logical file numbers	
$0513-$051C	1299-1308	Primary device numbers	
$051D-$0526	1309-1318	Secondary addresses	
$0527-$0530	1319-1328	IRQ Keyboard buffer	
407F8	2040	Monitor control for RAM/ROM	
$1000	4096	Start of Basic Text	
$4000	16384	Start of Basic Text when HIRES is being used.	

which key you are pressing, carefully type:

GRAPHIC0 [RETURN]

and you will return to the normal screen.

So we see that the operating system takes either $1000 (approx 4K) at the bottom of RAM for normal Basic or $4000 (approx 16K) for high resolution graphics.

What about the top of RAM? Well, apart from a small area from $FD00 to $FFFF which is again used by the system, the rest of RAM is available for Basic programs. The amount of memory from $1000 to $FCFF is 60671 and this is the number that appears on the screen at first power up.

However, to work all the Basic system of the computer, it needs another 32K of memory – the ROM. Where does this go – we already know that the computer can only read 64K of memory and it looks as though this is all taken by RAM. What happens is that the ROM for the operating system has addresses from $8000 to $FFF. So there are two different bytes of memory that have the same address one byte in RAM and another byte in ROM. When the computer is working it needs a switch between RAM and ROM so that the correct byte is read.

Different areas of memory are called memory banks and we need banking routines to switch different banks in or out.

The Plus/4 memory map may be illustrated by Diagram 1.

Diagram 1

Memory Area	RAM	ROM
$FD00 - $FFFF	Banking Routines	Operating Routines
$8000 - $FCFF	Available for Basic	Operating Routines
$4000 - $7FFF	BASIC AREA	
$1000 - $3FFF	Either BASIC or HIGH RES. GRAPHICS	
$0000 - $0FFF	RAM used by system	

When first switched on the computer has access to RAM memory bank $0000 to $7FFF and ROM $8000 to $FFFF. However the PEEK and POKE commands will always access RAM. A machine code program could be POKED to the top of RAM (below $FD00) but a system call to this part of memory would result in a ROM routine being entered at an unusual point and execution continued from this point. (Execution usually means death of the computer in this case – it will probably hang!)

The inbuilt monitor TEDMON allows you to display memory locations or disassemble either RAM or ROM.

Let us try some examples. Type and enter this Basic program:

```
10 SA=992
20 FORI=0to9
30 READX:POKE(SA+I),X
40 NEXT
50 DATA 162,26,138,157,200,12,
202,208,249,96
```

This is the type of program that appears in this and other computer magazines. The numbers in the data statement are POKED into memory one by one starting at location 992 if the program is RUN.

The simple machine code program following will be entered at $03E0 which is in the Tape Buffer area and thus safe from corruption.

```
03E0 A2 1A      LDX #$1A
03E2 8A         TXA
03E3 9D C8 0C   STA $0CC8,X
03E6 CA         DEX
03E7 D0 F9      BNE $03E2
03E9 60         RTS
```

Let us look at this in detail. The first command loads or sets the X register to $1A which is the same as 26, the number of letters in the alphabet. The command TXA transfers the value of X, which is 26, to the accumulator so both X and A are now 26. The location $0CC8 is the initial or zero column of the sixth row in the screen memory. The instruction STA $0CC8,X stores A in location $0CC8 plus the value of X which will be column 27 of row six (27 because the initial position corresponds to X = 0). The STA instruction is like a POKE so a letter Z will appear on the screen. The next line decrements X by one which now becomes 25. This is not zero so the BNE (Branch Not Equal) sends the processor back to the TXA. This will result in a Y being printed and so on until X is zero when the program will go to the RTS (ReTurn from Sub-routine) and return to Basic. So all the letters of the alphabet will be printed in reverse order. SYS 992 [RETURn] will run the routine.

OK, so far so good. Now let's try the Monitor. Type MONITOR and [RETURN]. Actually M and (shifted O) is an abbreviation that could be used.

Then type:

D 03E0 03E9

when the above machine code should be displayed.

Let us try to move this to an address where we have both ROM and RAM – above $8000.

Type X and [RETURN] to return to Basic. List the program already entered and change line 10 to read SA=32768. It is a good idea to save this program now. The number 32768 is the decimal equivalent of $8000, now RUN the program. The machine code will now be in RAM from $8000 and it might seem that SYS 32768 would run the program. Well try it! What happens is that the SYS call goes to the memory bank that is switched in, which is ROM above $8000 and happens to have a routine to give a warm start. If you saved your program then reload or type it out again. RUN to make sure the machine code is in RAM.

To check, go back into the monitor by typing M (Shifted O) and [RETURN] then D 8000 [RETURN].

What you now see is ROM and not our little program.

However the Monitor is controlled by location $07F8.

Type M 07F8 07F9 to display memory locations and type over the first 00 with 80 then [RETURN]. The monitor will now display RAM above $8000. Type D 8000 again and there should be our little machine code program. You should note that although the monitor is displaying RAM, if you used the G command, i.e. G 8000, then ROM would again be executed. (Don't do it!)

How can a machine code above $8000 be executed? The secret lies in two memory locations.

$FF3E — ROM select
$FF3F — RAM select

Any write (i.e. POKE) to $FF3E will select ROM and any write to $FF3F will select RAM. However care is needed. If you do POKE to $FF3F the machine will hang! This is because at the first interrupt the interrupt vector will point to RAM and not the usual interrupt routines in ROM. Thus the interrupt must be disabled before a switch to RAM and then cleared after the call to RAM is finished.

Type the following extra lines:

```
60 SA=992
70 FORI=0 to 11
80 READX:POKE(SA+I),X
90 NEXT
100 DATA 120,141,63,255,32,
0,128,141,62,255,88,96
```

Now, if you RUN the program in addition to entering the old routine at $8000 in RAM you will also have entered the following routine at $03E0:

```
03E0 78         SEI
03E1 8D 3F FF   STA $FF3F
03E4 20 00 80   JSR $8000
03E7 8D 3E FF   STA $FF3E
03EA 58         CLI
03EB 60         RTS
```

You could check by going into monitor and disassembling from $03E0 and also $8000. The command SYS 992 [RETURN] will run routine.

Let's examine this in detail. When you enter the command SYS 992 the program will go to $03E0 and set the interrupt i.e. the interrupt no longer occurs. The next op-code at $03E1 looks as though we are trying to put the value of A into the location $FF3F but actually this acts as a switch which changes from ROM to RAM. The next op-code makes the processor jump to the sub routine in RAM at $8000 and executes it – printing out the alphabet as before. The RTS at the end of the $8000 sub-routine returns the processor to $03E7 where the STA $FF3E switches back to ROM. The interrupt is returned to normal and the final RTS returns us to Basic.

GRAPHICALLY SPEAKING

Stuart Cooke takes a close look at Vidcom 64, a new low priced art package.

NO MATTER WHAT SORT OF PROGram you are writing for your C64, presentation is important. If you are writing a business package then the screen should be made to look as interesting as possible. On the other hand if you are writing a game you will need to provide an interesting backdrop for your game. There's no point in writing the best ever platform game if it isn't pleasing to the eye.

Designing screens on the C64 is not that easy since there are no inbuilt graphics commands available. Therefore, many programmers will use a graphics package that will help them draw 'pictures' in as short a time as possible and with ease. Numerous packages and peripherals are available. For example you could use a light pen or a touch tablet or even your joystick.

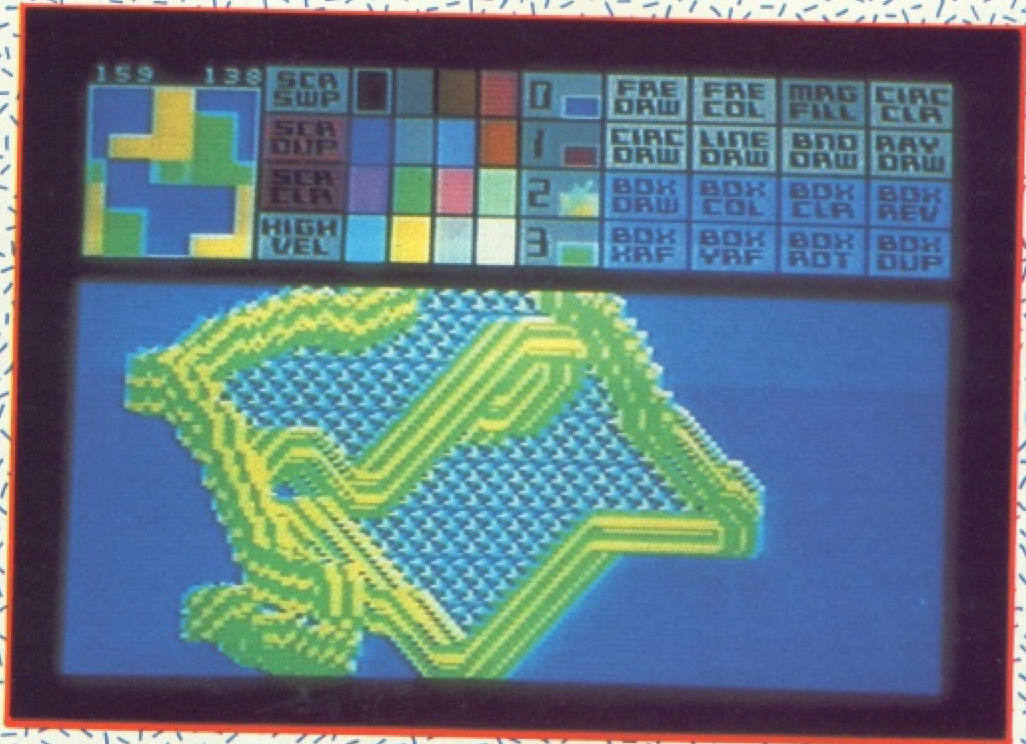
Vidcom 64 is another package to add to that already overflowing number of programs available, it does however have one feature that will make it stand way above the others, its price is only £4.95.

Vidcom's 64 will work on either the C64's multi colour or standard bit map screens. Standard bit map mode allows you to use two different colours in any character square on the screen while multi-colour mode allows you to have four colours in any square with a loss in horizontal resolution, i.e. the coloured 'dots' are twice as big.

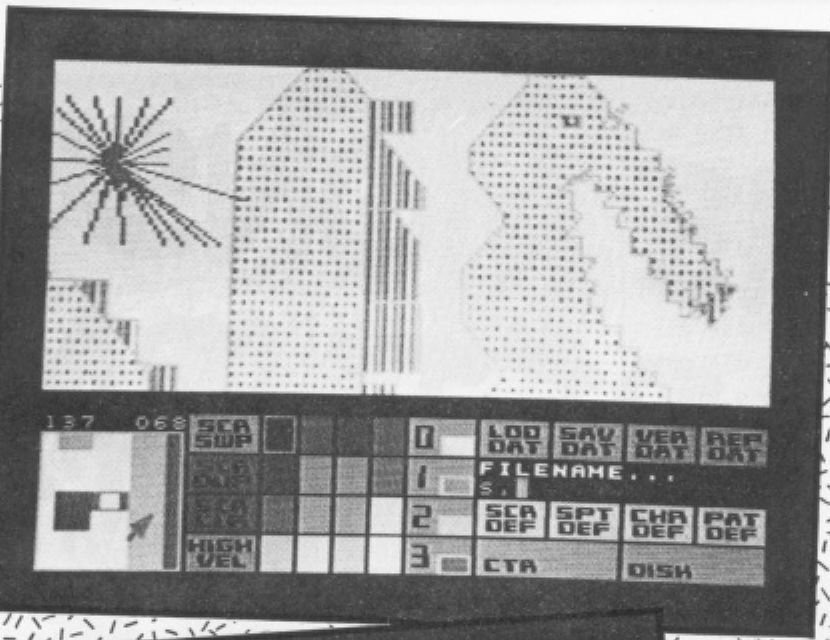
The best controller to use with Vidcom 64 is a trackball, however for those who can't afford one of these fairly expensive devices, a normal joystick will work just as well.

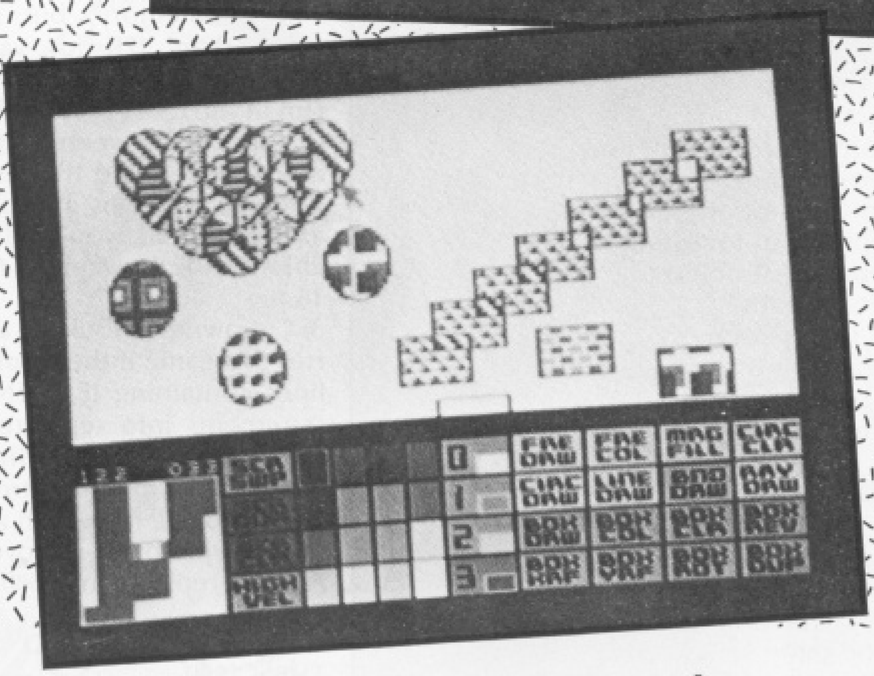
Layout

On entering the program the screen is split into two halves. The upper half is the top half of the screen on which you will draw, the other half is the control









panel which displays most of the functions available. I say most as there are actually three menus which appear at this position, each one being selected by the function keys. Moving your controller will move a small arrow around the screen allowing you to choose which command you want to use. Each press of the firebutton is acknowledged by a tone and the command that you have selected starts to flash.

Entering the drawing area is simple, you just have to press the control key. As I have previously said you can only see half of the drawing screen at once. However, if you move your pointer down the screen the command menu will flip up to the top of the screen allowing you to alter the contents of the bottom half of the screen. Sometimes this is quite difficult to use as you can't see what is on the other half of the screen to which you are writing on. Thankfully the author of the program has included a way of turning off the command menu allowing access to the full screen.

The three command menus available are the drawing menu which allows you to select all of the drawing commands, the definition menu which allows you to manipulate sprites, characters and patterns and the I/O menu which allows you to save all of your work to disk or tape.

Commands

All of the expected drawing commands are available. We have the freehand draw, band draw, box, circle etc. There are even some extra ones such as the ability to duplicate areas of the screen, rotate the contents of a box on the screen and reverse the contents of a box.

A few fill commands are also available from the drawing menu. Vidcom 64 has two graphic screens available for use. The-merge fill routines available in this menu allow you to merge areas of one of the graphics screens with the other.

A 'normal' fill routine is available from the definition.

The PAT FILL command allows you to fill areas of the screen with a pattern of your choice. This can be one of the patterns included in the program which consist of everything from a solid to a brick wall, or you can use the grab definition command which allows you to generate a new pattern by grabbing an area of the screen.

As you are no doubt aware, the screens in most games programs are not saved as a picture. Rather, the screens are built up from a number of redefined characters. The reason for this is that using characters will take up a lot less memory and you can use characters from one screen on another, saving even more space. With Vidcom 64 it is possible to grab character definitions from the screen. This means that you can use this package to design your screen and then save it as a number of characters. You can use these characters to design your other screens.

Not only can you grab characters but it is also possible to grab sprite definitions. Now it is a simple matter to change an area of the screen into sprites for use in another program.

Input/Output

The I/O menu offers a wide range of functions. It is possible to LOAD, SAVE, VERIFY and REPLACE data. Data can be the actual graphics screen, or the pattern, character or sprite definitions. It is also possible to select whether you are using cassette or disk from this menu.

Gripes

Obviously no program is perfect and I did think of a few improvements that could have been made to this package to really make it stand out from the rest.

There is no function that allows you to get a printer dump of the screen that you are designing. A screen dump is often quite useful for reference without having to load the screen back into the computer. There are many similar packages that do offer this facility.

I previously mentioned that the program makes a beeping noise whenever you select a command from one of the menus. However some of the drawing commands need more than one press of the fire button to operate. No indication is given when the program has acknowledged the first press. From experience this quite often means that you end up with circles and boxes that you can't see. A simple beep after each press of the fire button would have made things a lot clearer.

For £4.95 it is very difficult to fault Vidcom 64. It is an extremely easy to use and powerful program offering many facilities that are only available on more expensive programs.

If you don't own a graphics program then I would suggest that you go out and buy this. Even if you can't draw its great fun just messing around.

If you already own a graphics package then this is still worth looking at as it has some very powerful features.

Touchline

Name: Vidcom 64
Supplier: Activision
Address: 23, Pond St, Hampstead NW3 2PN
Price: £4.95

THE MONSTER RETURNS

Steve Lucas brings you an adventure program for the Plus/4.

THERE HAVE BEEN MANY rumours about my great uncle Victor Frankenstein having created a monster, but I have always dismissed them as superstitious nonsense. Imagine my surprise when I received a letter in the post from a solicitor informing me that Uncle Victor had died and asking me to come down to his mansion. With no clear idea of the reason for my journey, I travelled overnight to the lonely village where Uncle Victor's mansion stands and at this moment I am standing on the steps with just a note and a key to the house.

I have to discover the reason for my journey and solve this mystery. You should give me instructions as to what to do by typing in instructions in the form of one or two word sentences.

Here is a list of some of the words I understand: go, in, out, up, down, n, s, e, w, help, look, search, open, unlock, kill, attack, sew, fit, insert, cut, dig, score, rub, ride, screw, unscrew, get, take, drop, leave, put, pull, connect, drink, eat, wait, swim, wash

Notes

1. Line 10 forces the computer to use the upper/lower case mode and therefore all instructions must be given to the computer in lower case only.

2. The listing takes up just over 14K of memory, but needs more than 17K for variable storage. It will not, therefore run in the C-16. If, however, all the spaces between keywords are removed and the descriptions of locations and objects are shortened, you should be able to run the game in the C-16.

3. C64 owners should be able to run this game if they: Split any lines containing IF THEN ELSE statements into separate lines with different conditions; replace the main control DO LOOP UNTIL loop with a conditional jump ie. remove line 790 and replace line 1580 with:

1580 IF (P%=24 AND AT=2) THEN 1590
1581 GOTO 800

Replace scnclr with a PRINT "clear screen" command; GETKEY A$ just waits for a key to be pressed and can be replaced with:

770 GET A$: IF A$="" THEN 770

Variables Used

S%(X,Y)	holds the map
e,f	tests action
k	check for word recognition
P%	current location
B%(x,y)	pointer to location of objects
Q$(x)	descriptions of locations
G$(x)	descriptions of objects
aa-az	flags for actions
N$(x)	words understood
N%(x)	pointer to words
X$	input sentence
C$,B$,D$,L$	substrings of input sentence

Program Breakdown

10	selects lower case/upper case mode
20-130	instructions
140-190	fills arrays and set variables
200-750	data for game
760-780	initialise
790-1580	main control loop..do loop until win game
800-860	test for traps etc
870	describe location
880-1040	directions
1050-1090	describe objects
1100	input sentence
1120-1570	call appropriate subroutine (if necessary!)
1590	win game
1600-3570	subroutines

```
10 PRINT CHR$(14);CHR$(9)
20 PRINT"[CLEAR][RED]"
30 CHAR 1,9,3,"[s T]HE [s M]
ONSTER [s R]ETURNS"
40 CHAR 1,9,4,"==============
======"
50 PRINT:PRINT:PRINT"[BLUE][
s M]Y NAME IS [s H]ERBERT [s
 F]RANKENSTEIN AND [s I] "
60 PRINT"HAVE JUST RECIEVED
A NOTE IN THE POST"
70 PRINT"FROM MY UNCLE, [s V
]ICTOR [s F]RANKENSTEIN WHO
"
80 PRINT"DIED UNDER MYSTERIO
US CIRCUMSTANCES LASTWEEK."
90 PRINT"[s T]HE NOTE READS
:-"
100 PRINT:PRINT"[GREEN][s Y]
OU MUST DESTROY THE MONSTER
BEFORE IT  IS TOO LATE !!!!"
110 PRINT:PRINT"[BLUE][s I]
AM AT THIS MOMENT STANDING O
N THE"
120 PRINT"[s T]HE STEPS OF U
NCLE [s V]ICTOR'S MANSION."
130 PRINT:PRINT:PRINT"[RED][
s P]LEASE HELP ME IN MY TASK
..."
140 DIM Q$(60),B%(40,1),V$(1
0),G$(50),N%(40),N$(50),X$(5
0),S%(60,4)
150 FOR X=1 TO 26:READ G$(X)
,B%(X,1),N$(X):N%(X)=X:NEXT
160 FOR X=1 TO 46:READ Q$(X)
:FOR Y=1 TO 4:READ S%(X,Y):N
EXT Y,X
170 P%=13:N$(25)="CROWBAR":G
$(20)="":G$(21)=""
180 B%(25,1)=29:G$(25)="A ST
RONG CROWBAR":G$(3)="A NEEDL
E AND THREAD"
190 N$(3)="NEEDLE"
200 DATA A TATTY NOTE,13,NOT
E,AN OLD BRASS KEY,13,KEY,A
REEL OF COTTON
210 DATA 4,COTTON,A LEMON,4,
LEMON,A PLASTIC DUSTBIN,1,DU
STBIN
220 DATA A CANDLE,3,CANDLE,A
 GOLD PEN IN A HOLDER,6,PEN
230 DATA A MONSTER STRAPPED
TO A TABLE,19,MONSTER,A SCAL
PEL,20,SCALPEL
240 DATA A LARGE GLASS BOTTL
E,21,BOTTLE,A SMALL LEVER,22
,LEVER
250 DATA A PADLOCK,22,PADLOC
K,A PAINTING ON THE WALL,8,P
AINTING
260 DATA A PAIR OF SLIPPERS,
5,SLIPPERS,"",8,MAP
270 DATA AN OLD LEATHER DIAR
Y,12,DIARY,A STALE SANDWICH,
9,SANDWICH,A SPADE,28
280 DATA SPADE,A COFFIN,41,C
```



```
OFFIN,A CORPSE,41,CORPSE,A H
EART,41,HEART
290 DATA A SCREWDRIVER,39,SC
REWDRIVER,A PAIR OF ELECTROD
ES,46,ELECTRODES
300 DATA A CRUCIFIX,30,CRUCI
FIX,"",8,MAP,"",1,MATCH
310 DATA"IN THE BACKYARD. [s
 A] PILE OF BOXES STANDS IN
ONE CORNER.",0,0,0,0
320 DATA"IN A COMFORTABLE LO
UNGE.",0,0,3,0
330 DATA"IN THE DINING ROOM.
 [s A] HIGHLY POLISHED   TAB
LE STANDS IN THE CENTRE."
340 DATA 0,6,4,2,"IN THE KIT
CHEN. [s T]HE SINK IS FULL O
F     DIRTY POTS",0,7,0,3
350 DATA"IN A BEDROOM. [s T]
HERE IS A LARGE FOUR     POS
TER BED HERE.",0,8,0,0
360 DATA"IN A LARGE OAK PANE
LLED LIBRARY.",3,0,7,0
370 DATA"IN THE HALL. [s A]
WIDE STAIRCASE LEADS UP  FRO
M HERE.",4,13,0,0
380 DATA"ON THE LANDING. [s
A] STAIRCASE LEADS DOWN.",5,
12,9,0
390 DATA"IN A SMALL BEDROOM.
",0,0,10,8
400 DATA"IN A CLOSET FULL OF
 OLD CLOTHES. [s A] RAT  LIE
S IN ONE CORNER.",0,0,0,9
410 DATA"IN THE TOILET.",0,0
,0,12
420 DATA"IN THE BATHROOM. [s
 A] TAP DRIPS INTO THE   RUS
TY BATH.",8,0,11,0
430 DATA"ON THE DOORSTEP OF
AN OLD MANSION.",0,14,0,0
440 DATA"ON A DIRT TRACK LEA
DING THROUGH AN      OVERGRO
WN GARDEN.",13,23,0,0
450 DATA"IN A SECRET PASSAGE
.",0,16,6,0,"AT THE TOP OF A
 FLIGHT OF STAIRS."
460 DATA 15,0,0,0,"AT THE BO
TTOM OF A FLIGHT OF STEPS. [
s A]   PASSAGE LEADS EAST."
470 DATA 0,0,18,0,"AT THE NO
RTHERN END OF A SECRET
  LABORATORY.",0,19,0
480 DATA 17,"IN A SECRET LAB
ORATORY. [s T]HERE IS AN
 OPERATING TABLE HERE."
490 DATA 18,20,0,0,"IN A PRE
P ROOM. [s T]HERE IS A GLASS
 CABINETON THE WALL.",19,22
500 DATA 21,0,"IN A CHEMICAL
 STORE ROOM.",0,0,0,20
510 DATA"IN A SMALL ROOM. [s
 T]HERE IS A LARGE        CON
TROL PANEL HERE.",20,0,0,0
520 DATA"ON A DIRT TRACK. [s
 T]HERE IS A SIGNPOST     HER
E.",14,33,24,0
530 DATA"ON A SECRET FOOTPAT
H LEADING THROUGH A  PEAT BO
G.",0,0,25,23
540 DATA"STANDING OUTSIDE A
HUT WHICH IS        SURROUN
DED BY TREES.",0,0,0,24
550 DATA"AT ONE END OF THE H
UT.",0,27,0,0
560 DATA"INSIDE A SMALL HUT.
 [s A] TABLE STANDS        AGA
INST THE WALL.",26,0,29,28
570 DATA"IN THE HUT. [s T]HE
RE'S A BAG HERE.",0,0,27,0
580 DATA"IN THE HUT.",0,0,0,
27
590 DATA"BY AN ENORMOUS ALTA
R. [s A]NCIENT SEPULCRES LIN
E THE WALLS.",0,31,0,0
600 DATA"INSIDE A LARGE CRYP
T.",30,32,0,0
610 DATA"OUTSIDE A CRYPT. [s
 M]IST SWIRLS AROUND THE ENT
RANCE.",0,0,33,44
620 DATA"ON A DIRT TRACK.",2
3,34,0,32
630 DATA"BY A LARGE METAL GA
TE.",33,35,0,0
640 DATA"ON A DIRT TRACK. [s
 T]HE WAY SOUTH LEADS     INT
O A GLOOMY FOREST.",34,36
650 DATA 0,0,"IN A GLOOMY FO
REST.",35,37,40,38
660 DATA"IN A SMALL CLEARING
. [s T]HE FOREST IS TOO  THI
CK TO TRAVEL FURTHER.",36
670 DATA 0,0,0,"LOST IN A GL
OOMY FOREST.",38,38,38,38
680 DATA"INSIDE THE FORESTER
'S MEAGRE COTTAGE.",0,0,0,0
690 DATA"OUTSIDE A WOODCUTTE
R'S COTTAGE.",0,0,0,36
700 DATA"AT THE BOTTOM OF A
FRESHLY DUG GRAVE.",0,0,0,0
710 DATA"BY A FRESHLY DUG GR
AVE.",0,0,43,34
720 DATA"BY SOME TOMBSTONES.
",0,0,0,42
730 DATA"ON A GRAVEL PATH.",
0,45,32,0
740 DATA"BY A GRANITE MONOLI
TH. [s A] LARGE RED LEVERPRO
TRUDES FROM THE BASE."
750 DATA 44,0,0,0,"INSIDE A
SECRET CHAMBER.",45,0,0,0
760 PRINT"[DOWN][DOWN]   [RV
SON][BLUE]  [s P]RESS ANY KE
Y TO START THE GAME."
770 GETKEY A$
780 SCNCLR:S%(7,4)=6
790 DO
800 IF AT=1 AND P%=19 THEN A
T=2:PRINT"[s T]HE MONSTER SE
ES ME. [s I]'D BETTER MOVE I
T !!!"
810 IF AT=2 AND AX=1 THEN X$
="[s T]HE MONSTER CATCHES ME
!":GOSUB 2000
820 IF AT=2 THEN PRINT"[s T]
HE MONSTER IS JUST ONE STEP
BEHIND ME!"
830 IF P%=15 AND AE<>2 THEN
X$="[s I] TRIPPED AND FELL I
N THE DARK!":GOSUB 2000
840 IF P%=19 AND AR=2 THEN P
RINT"[s T]HE MONSTER HAS A P
AIR OF ELECTRODES    ATTACHE
D!"
850 IF P%=34 AND AK<2 THEN P
RINT"[s A] VICIOUS WOLF GROW
LS AT ME!"
860 IF P%=24 AND AJ<3 THEN X
$="[s I] SINK INTO THE BOG!"
:GOSUB 2000
870 PRINT"[DOWN][DOWN][BLUE]
[s I] AM :-"
880 PRINT"[DOWN][RED]";Q$(P%
):K=0:A$="":A%=P%
890 IF S%(P%,1)>0 THEN A$="[
s N]ORTH"
900 IF S%(P%,2)>0 AND LEN(A$
)>0 THEN A$=A$+",[s S]OUTH"
910 IF S%(P%,2)>0 AND LEN(A$
)=0 THEN A$="[s S]OUTH"
920 IF S%(P%,3)>0 AND LEN(A$
)>0 THEN A$=A$+",[s E]AST"
930 IF S%(P%,3)>0 AND LEN(A$
)=0 THEN A$="[s E]AST"
940 IF S%(P%,4)>0 AND LEN(A$
)>0 THEN A$=A$+",[s W]EST"
950 IF S%(P%,4)>0 AND LEN(A$
)=0 THEN A$="[s W]EST"
960 IF P%=1 THEN A$="[s I]N"
970 IF P%=4 OR P%=31 OR P%=2
```

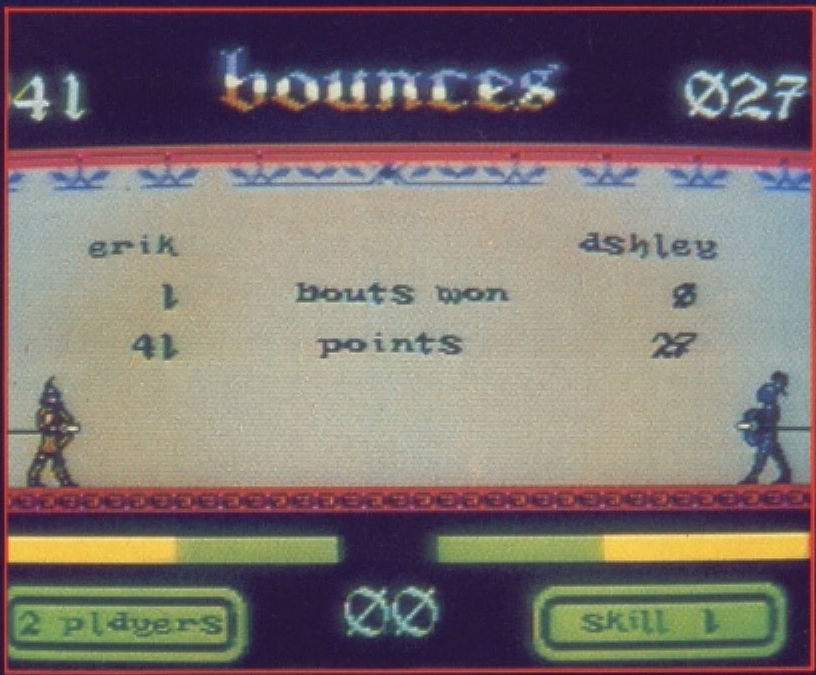
```
6 THEN A$=A$+",[s O]UT"
980 IF (P%=25 OR P%=40 OR P%
=32) THEN A$=A$+",[s I]N"
990 IF P%=39 THEN A$="[s O]U
T"
1000 IF P%=16 OR P%=8 THEN A
$=A$+",[s D]OWN"
1010 IF P%=7 OR P%=17 OR P%=
41 THEN A$=A$+" [s U]P"
1020 IF P%=42 AND AM>1 THEN
A$=A$+", [s D]OWN"
1030 PRINT"[DOWN][DOWN][BLUE
][s I] CAN GO :-"
1040 PRINT"[RED]";A$
1050 E=0:FOR X=1 TO 26:PP%=0
:IF B%(X,1)=P% THEN PP%=1
1060 IF PP%=1 THEN 1080
1070 NEXT:GOTO 1100
1080 IF E=0 THEN PRINT"[BLUE
][DOWN][s I] CAN SEE :-[RED]
"
1090 PRINTG$(X):E=1:GOTO 107
0
1100 PRINT"[DOWN][BLUE][s W]
HAT SHALL [s I] DO NOW [PURP
LE] ";:INPUT Z$
1110 B$=LEFT$(Z$,2):C$=LEFT$
(Z$,3):D$=LEFT$(Z$,4)
1120 SCNCLR:SOUND 1,800,1
1130 IF C$="PRA" THEN PRINT"
[BLACK][s P]RAYING MAKES ME
FEEL BETTER BUT NOTHING HAPP
ENS.":K=1
1140 IF C$="HEL" THEN PRINT"
[BLACK][s I]'M SORRY [s I] H
AVEN'T A CLUE!":K=1
1150 IF RIGHT$(Z$,3)="OFF" T
HEN K=1:PRINT"[s M]IND YOUR
LANGUAGE!"
1160 IF C$="SCO" THEN K=1:PR
INT"[s T]HIS IS NO GAME YOU
KNOW!"
1170 IF C$="RUB" THEN K=1:PR
INT"[s D]ON'T BE ABSURD!"
1180 IF C$="RID" THEN K=1:PR
INT"[s W]HAT IN ?"
1190 IF C$="INV" THEN K=1:GO
SUB 2070
1200 IF C$="REA" THEN K=1:GO
SUB 2100
1210 IF C$="OPE" THEN K=1:GO
SUB 2260
1220 IF C$="UNS" OR C$="SCR"
 THEN K=1:GOSUB 2310
1230 IF C$="UNL" THEN K=1:GO
SUB 2360
1240 IF C$="DRO" OR C$="LEA"
 OR C$="PUT" THEN K=1:GOSUB
2400
1250 IF C$="IN" OR D$="GO I"
 THEN K=1:GOSUB 2660
1260 IF C$="OUT" OR D$="GO O
" THEN K=1:GOSUB 2730
1270 IF C$="UP" OR B$="U" OR
 D$="GO U" THEN K=1:GOSUB 27
90
1280 IF C$="PHO" THEN K=1:PR
INT"[s W]HAT WITH ?"
1290 IF C$="DOW" OR B$="D" O
R D$="GO D" THEN K=1:GOSUB 2
830
1300 IF C$="SEA" THEN K=1:GO
SUB 2880
1310 IF C$="LIG" THEN K=1:GO
SUB 2920
1320 IF C$="SQU" THEN K=1:GO
SUB 2970
1330 IF C$="DIA" THEN K=1:GO
SUB 3040
1340 IF C$="DIG" THEN K=1:GO
SUB 3150
1350 IF C$="SMA" OR C$="BRE"
 THEN K=1:GOSUB 3180
1360 IF C$="PUL" THEN K=1:GO
SUB 3220
1370 IF C$="ATT" OR C$="KIL"
 THEN K=1:GOSUB 3310
1380 IF C$="LOO" THEN K=1:PR
INT"[s I] CAN'T SEE ANYTHING
 SPECIAL!"
1390 IF C$="CUT" OR C$="STA"
 THEN K=1:GOSUB 3340
1400 IF C$="FIT" OR C$="INS"
 THEN K=1:GOSUB 3390
1410 IF C$="SEW" THEN K=1:GO
SUB 3460
1420 IF C$="CON" THEN K=1:GO
SUB 3500
1430 IF C$="TAL" OR C$="SAY"
 THEN K=1:PRINT"[s I] DON'T
TALK TO MYSELF!"
1440 IF C$="WAI" THEN K=1:FO
R X=1 TO 1000:NEXT X
1450 IF P%=24 AND AT=2 THEN
GOSUB 3560
1460 IF C$="EAT" THEN PRINT"
[s I]'M NOT HUNGRY AT THE MO
MENT!":K=1
1470 IF C$="DRI" THEN PRINT"
[s I]'M NOT THIRSTY!":K=1
1480 IF (B$="N" OR D$="GO N"
) AND S%(P%,1)>0 THEN K=1:P%
=S%(P%,1)
1490 IF (B$="S" OR D$="GO S"
) AND S%(P%,2)>0 THEN K=1:P%
=S%(P%,2)
1500 IF (B$="E" OR D$="GO E"
) AND S%(P%,3)>0 THEN K=1:P%
=S%(P%,3)
1510 IF (B$="W" OR D$="GO W"
) AND S%(P%,4)>0 THEN K=1:P%
=S%(P%,4)
1520 IF K=0 AND (B$="N" OR B
$="S" OR B$="E" OR B$="W") T
HEN K=1:PRINT"[s I] CAN'T"
1530 IF C$="SWI" THEN K=1:PR
INT"[s D]ON'T BE ABSURD!"
1540 IF C$="WAS" THEN K=1:PR
INT"[s D]ON'T BE A SILLY BIL
LY!"
1550 IF C$="GET" OR C$="TAK"
 OR C$="GRA" THEN K=1:GOSUB
1600
1560 AX=0:IF P%=A% THEN AX=1
1570 IF K=0 THEN PRINT"[s I]
'M AFRAID [s I] DON'T UNDERS
TAND YOU"
1580 LOOP UNTIL (P%=24 AND A
T=2)
1590 GOSUB 3560
1600 GOSUB 1950
1610 IF L%<>1 THEN PRINT"[s
I] DON'T SEE A ";L$;" HERE":
RETURN
1620 E%=0:FOR X=1 TO 26
1630 IF B%(X,1)=P% AND B%(N%
(R),1)=P% THEN E%=1
1640 NEXT:IF E%=0 THEN PRINT
"[s M]AYBE [s I] NEED GLASSE
S, BUT [s I] DON'T SEE IT":R
ETURN
1650 IF R=1 THEN AA=1
1660 IF R=2 THEN AB=1
1670 IF R=3 AND AC=0 THEN AC
=1
1680 IF R=4 THEN AD=1
1690 IF R=6 THEN AE=1
1700 IF R=9 THEN AF=1
1710 IF R=13 AND P%=8 AND AG
=0 THEN AG=1:GOSUB 3120:RETU
RN
1720 IF R=7 AND AH=0 THEN AH
=1:PRINT"[s A] PANEL SLID AS
IDE":S%(6,4)=15
1730 IF R=26 THEN AI=1
1740 IF R=8 THEN X$="[s I]TS
 HANDS CLAMP AROUND MY THROA
T. [s A][s A]GGH":GOSUB 2000
1750 IF R=12 THEN PRINT"[s I
]T'S LOCKED AROUND THE LEVER
!":RETURN
1760 IF R=11 THEN PRINT"[s D
]ON'T BE A FOOL!":RETURN
1770 IF R=15 AND AJ=0 THEN A
J=1
1780 IF R=17 AND AK=0 THEN A
K=1
1790 IF R=17 AND AK=2 THEN R
ETURN
1800 IF R=18 THEN AM=1
1810 IF R=22 THEN AN=1
1820 IF R=19 THEN PRINT"[s I
]T'S TOO HEAVY!":RETURN
1830 IF R=20 THEN PRINT"[s I
] CAN'T CARRY IT!":RETURN
1840 IF R=21 AND AN<3 THEN P
RINT"[s I] CAN'T DO THAT....
YET!":RETURN
1850 IF R=16 THEN AO=1
1860 IF R=25 THEN AP=1
1870 IF R=23 AND AR=0 THEN A
R=1
1880 IF (R=21 AND AS=0) THEN
 AS=1
1890 E%=0:FOR X=1 TO 5
1900 IF V$(X)="" THEN V$(X)=
G$(N%(R)):E%=1:X=[111]
1910 NEXT
1920 IF E%=0 THEN PRINT"[s S
]ORRY...MY HANDS ARE FULL!!!
":RETURN
1930 B%(N%(R),1)=0
1940 RETURN
1950 L$="":FOR X=1 TO LEN(Z$
)
1960 IF MID$(Z$,X,1)=" " THE
N L$=RIGHT$(Z$,(LEN(Z$)-X)):
X=99
1970 NEXT:R=X:L%=0:IF LEN(L$
)<2 THEN RETURN
1980 FOR X=1 TO 26:IF LEFT$(
N$(X),LEN(L$))=L$ THEN L%=1:
R=X
1990 NEXT:RETURN
2000 PRINT"[CLEAR][DOWN][DOW
N]";X$
2010 PRINT"[DOWN][DOWN][DOWN
][RED] [s I] AM DEAD"
2020 PRINT"[DOWN][DOWN][DOWN
]  [BLUE][RVSON] [s W]OULD Y
OU LIKE ANOTHER GAME <Y/N> ?
 "
2030 GETKEY A$
2040 IF A$="Y" OR A$="[s Y]"
 THEN RUN
2050 IF A$="N" OR A$="[s N]"
 THEN PRINT"[CLEAR][s G]OODB
YE. [s T]HANK YOU FOR PLAYIN
G!":END
2060 GOTO 2030
2070 PRINT"[PURPLE][s I] AM
CARRYING :-":F=0:FOR X=1 TO
5:IF V$(X)<>"" THEN F=1:PRIN
TV$(X)
2080 NEXT:IF F=0 THEN PRINT"
[s N]OTHING AT ALL!"
```

```
2090 RETURN
2100 GOSUB 1950
2110 IF P%=23 THEN PRINT"[s
T]HE SIGNPOST READS :-"
2120 IF P%=23 THEN PRINT"[BL
ACK][s B][s E][s W][s A][s R
][s E] [s O][s F] [s T][s H]
[s E] [s B][s O][s G]!!!!!"
2130 IF P%=23 AND AJ=1 THEN
PRINT"[s I] THINK [s I] CAN
FOLLOW THE MAP":AJ=3
2140 IF P%=23 THEN RETURN
2150 IF R=16 AND AO=1 THEN P
RINT"[s T]HE DIARY READS :-"
:GOSUB 2220:RETURN
2160 IF P%=45 THEN X$="A SUR
GE OF POWER RUNS THROUGH MY
BODY":GOSUB 2000
2170 IF AA<>1 THEN PRINT"[s
I] CAN'T DO THAT JUST YET":R
ETURN
2180 PRINT"[s T]HE NOTE READ
S :-"
2190 PRINT"[s T]HERE'S NOT M
UCH POINT IN THAT!":RETURN
2200 PRINT"[RED][s D][s E][s
 S][s T][s R][s O][s Y] [s T
][s H][s E] [s M][s O][s N][
s S][s T][s E][s R]"
2210 RETURN
2220 PRINT"[BLACK][s A]PRIL
1ST 1895"
2230 PRINT"[s T]HE MONSTER M
UST BE DESTROYED!!"
2240 PRINT"[s F]IND THE ELEC
TRODES AND BRING IT BACK TOL
IFE!"
2250 PRINT"[s T][s H][s E][s
 N] [s T][s A][s K][s E] [s
I][s T] [s T][s O] [s T][s H
][s E] [s B][s O][s G]!!!!":
RETURN
2260 IF P%=41 THEN GOSUB 231
0:RETURN
2270 IF P%=8 AND AG>0 THEN P
RINT"[s I]'D BETTER [s D][s
I][s A][s L] THE COMBINATION
!":RETURN
2280 IF P%<>13 THEN PRINT"[s
 N]OT HERE!":RETURN
2290 IF AB<2 THEN PRINT"[s I
]T'S LOCKED!":RETURN
2300 PRINT"[s O].[s K].":S%(
13,1)=7:RETURN
2310 IF P%<>41 THEN PRINT"[s
 I] CAN'T DO THAT HERE!":RET
URN
2320 IF AN<1 THEN PRINT"[s I
] HAVE NO SCREWDRIVER!":RETU
RN
2330 IF AN>1 THEN PRINT"[s I
]T'S ALREADY OPEN!":RETURN
2340 PRINT"[s I] OPEN THE CO
FFIN!":G$(20)="A CORPSE"
2350 G$(19)="AN OPEN COFFIN"
:AN=2:RETURN
2360 IF P%=8 AND AG>0 THEN P
RINT"[s I]'D [s D][s I][s A]
[s L] THE COMBINATION!":RETU
RN
2370 IF P%<>13 THEN PRINT"[s
 N]OT HERE!":RETURN
2380 IF AB<>1 THEN PRINT"[s
I] NEED THE KEY!":RETURN
2390 PRINT"[s O].[s K].":AB=
2:RETURN
2400 GOSUB 1950
2410 IF L%<>1 THEN PRINT"[s
I] DON'T SEE A ";L$
2420 E=0:FOR X=1 TO 5:IF V$(
X)=G$(N%(R)) THEN V$(X)="":E
=1
2430 NEXT:IF E=0 THEN PRINT"
[s I] DON'T HAVE IT!":RETURN
2440 B%(N%(R),1)=P%
2450 IF R=1 THEN AA=0
2460 IF R=2 AND AB=1 THEN AB
=0
2470 IF R=3 AND AC=1 THEN AC
=0
2480 IF R=4 THEN AD=0
2490 IF R=6 THEN AE=0
2500 IF R=9 THEN AF=0
2510 IF R=10 THEN X$="[s T]H
E BOTTLE BREAKS AND SPRAYS M
E WITH    ACID!":GOSUB 1190
2520 IF R=26 THEN AI=0
2530 IF R=15 AND AJ=1 THEN A
J=0
2540 IF R=17 AND AK=1 AND P%
=34 THEN AK=2:S%(34,3)=42:GO
SUB 2630:RETURN
2550 IF R=17 THEN AK=0
2560 IF R=18 AND AM=1 THEN A
M=0
2570 IF R=22 AND AN=1 THEN A
N=0
2580 IF R=16 THEN AO=0
2590 IF R=25 THEN AP=0
2600 IF R=23 AND AR=1 THEN A
R=0
2610 IF R=21 AND AS=1 THEN A
S=0
2620 RETURN
2630 PRINT"[s T]HE WOLF EATS
 IT AND RUNS AWAY!"
2640 G$(17)="SOME CRUMBS":N$
(17)="CRUMBS"
2650 RETURN
2660 IF P%=13 AND AB<2 THEN
PRINT"[s T]HE DOOR'S LOCKED!
":RETURN
2670 IF P%=13 THEN P%=7:RETU
RN
2680 IF P%=1 THEN P%=4:RETUR
N
2690 IF P%=25 THEN P%=26:RET
URN
2700 IF P%=40 THEN P%=39:RET
URN
2710 IF P%=32 THEN P%=31:RET
URN
2720 PRINT"[s S]ORRY!":RETUR
N
2730 IF P%=26 THEN P%=25:RET
URN
2740 IF P%=39 THEN P%=40:RET
URN
2750 IF P%=31 THEN P%=32:RET
URN
2760 IF P%=4  THEN P%=1 :RET
URN
2770 IF P%=7  THEN P%=13:RET
URN
2780 PRINT"[s D]ON'T BE ABSU
RD!":RETURN
2790 IF P%=41 THEN P%=42:RET
URN
2800 IF P%=17 THEN P%=16:RET
URN
2810 IF P%=7 THEN P%=8:RETUR
N
2820 PRINT"[s D]ON'T BE STUP
ID!":RETURN
2830 IF P%=42 AND AM<2 THEN
PRINT"[s I]'D NEED TO DIG A
HOLE FIRST!":RETURN
2840 IF P%=42 THEN P%=41:RET
URN
2850 IF P%=16 THEN P%=17:RET
URN
2860 IF P%=8 THEN P%=7:RETUR
N
2870 PRINT"[s D]ON'T BE SILL
Y!":RETURN
2880 IF P%<>1 THEN PRINT"[s
I] CAN'T SEE ANYTHING HERE!"
:RETURN
2890 IF AI=0 THEN PRINT"[s I
] SEE SOMETHING AMONGST THE
RUBBISH"
2900 IF AI=0 THEN G$(26)="A
BOX OF MATCHES":AI=1:RETURN
2910 PRINT"[s T]HERE'S NOTHI
NG ELSE HERE!":RETURN
2920 IF AE<1 THEN PRINT"[s H
]OW AM [s I] SUPPOSED TO DO
THAT ?":RETURN
2930 IF AI<1 THEN PRINT"[s I
] HAVEN'T GOT A LIGHT!":RETU
RN
2940 PRINT"[s O].K":AE=2
2950 FOR X=1 TO 5:IF V$(X)=G
$(6) THEN V$(X)=G$(6)+" (LIT
)":G$(6)=V$(X)
2960 NEXT:RETURN
2970 IF AD<>1 THEN PRINT"[s
I] HAVE NOTHING TO SQUEEZE":
RETURN
2980 IF AA<>1 THEN PRINT"SOM
E DROPS OF LEMON JUICE FALL
TO THE    FLOOR!":RETURN
2990 IF AL=0 THEN AL=2000+IN
T(RND(0)*2000)
3000 PRINT"SOME JUICE FALLS
ONTO THE NOTE AND A     SECRE
T MESSAGE APPEARS."
3010 PRINT"[s I]T READS :-"
3020 PRINT"[BLACK][s T]O OPE
N THE SAFE DIAL :-";AL
3030 RETURN
3040 IF P%<>8 THEN PRINT"NOT
 HERE!":RETURN
3050 IF AG<1 THEN PRINT"NOT
YET!":RETURN
3060 IF AL=0 THEN PRINT"[s I
] DON'T KNOW THE COMBINATION
!":RETURN
3070 INPUT"[s W]HAT DO [s I]
 DIAL ";BA
3080 IF AL<>BA THEN PRINT"[s
 W]RONG COMBINATION":RETURN
3090 PRINT"[s T]HE SAFE OPEN
S"
3100 G$(15)="AN OLD MAP SHOW
ING A SECRET PATH THROUGH TH
E BOG"
3110 RETURN
3120 Q$(8)=Q$(8)+" [s T]HERE
 IS A SAFE ON THE WALL"
3130 PRINT"[s T]HE PAINTING
SWINGS ASIDE ON HINGES TO  R
EVEAL A SAFE."
3140 RETURN
3150 IF AM<>1 THEN PRINT"[s
W]ITH MY BARE HANDS ?":RETUR
N
3160 IF P%<>42 THEN PRINT"[s
 T]HE GROUND'S TOO HARD!":RE
TURN
3170 PRINT"[s I] DIG DOWN TO
 THE COFFIN!":AM=2:RETURN
3180 IF AP<>1 THEN PRINT"[s
W]HAT WITH..?":RETURN
3190 IF P%<>22 THEN PRINT"[s
 N]OT HERE!":RETURN
```

```
3200 IF AQ<1 THEN AQ=1:PRINT"[s T]HAT'S DONE THE TRICK!":G$(12)="A BROKEN PADLOCK"
3210 RETURN
3220 IF P%=22 AND AQ<1 THEN PRINT"[s I]T'S LOCKED!":RETURN
3230 IF P%=22 AND AR<2 THEN PRINT"[s I] HAVEN'T CONNECTED THE ELECTRODES YET!":RETURN
3240 IF P%=22 AND AS<2 THEN PRINT"[s I] HAVEN'T FOUND A NEW HEART YET":RETURN
3250 IF P%=22 AND AC<3 THEN PRINT"[s I]'LL HAVE TO SEW IT UP FIRST!":RETURN
3260 IF P%=22 THEN AT=1:PRINT"[s T]HE MONSTER COMES TO LIFE...":G$(8)="AN EVIL MONSTER"
3270 IF P%=22 THEN RETURN
3280 IF P%=45 AND AU=0 THEN PRINT"[s S]OMETHING HAPPENED":S%(45,2)=46:AU=1:RETURN
3290 IF P%=45 THEN X$="A SURGE OF POWER FLOWS THROUGH MY BODY":GOSUB 2000
3300 PRINT"[s T]HERE'S NOT MUCH POINT IN THAT!":RETURN
3310 IF P%=19 THEN X$="[s T]HE MONSTER'S HANDS GRAB MY THROAT":GOSUB 2000
3320 IF P%=34 THEN X$="[s T]HE WOLF ATTACKS ME":GOSUB 2000
3330 PRINT"[s I] SEE NO POINT IN VIOLENCE HERE!":RETURN
3340 IF P%<>41 THEN PRINT"[s N]OT HERE!":RETURN
3350 IF AF<>1 THEN PRINT"[s W]HAT WITH ?":RETURN
3360 IF AN=3 THEN RETURN
3370 AN=3:PRINT"[s I]'LL HAVE TO CLOSE MY EYES TO DO THIS!"
3380 G$(20)="A MUTILATED CORPSE":G$(21)="A FRESH HEART":RETURN
3390 IF P%<>19 THEN PRINT"[s I] CAN'T DO THAT HERE!":RETURN
3400 IF AS=2 THEN RETURN
3410 IF AS=0 THEN PRINT"[s N]O HEART!":RETURN
3420 PRINT"[s I] FIT THE HEART..[s I]'LL NEED TO SEW IT UP NOW!"
3430 G$(8)=G$(8)+" (WITH A NEW HEART)"
3440 FOR X=1 TO 5:IF V$(X)=G$(21) THEN V$(X)=""
3450 NEXT:AS=2:RETURN
3460 IF AC<1 THEN PRINT"[s I] HAVE NO THREAD!":RETURN
3470 IF AC=3 THEN RETURN
3480 IF AS<2 THEN PRINT"[s I] HAVEN'T FITTED THE HEART YET!":RETURN
3490 AC=3:PRINT"[s O].[s K]. [s I] SEW IT UP":RETURN
3500 IF P%<>19 THEN PRINT"[s W]HAT AN ABSURD IDEA!":RETURN
3510 IF AR<1 THEN PRINT"[s H]OW ?":RETURN
3520 IF AR=2 THEN PRINT"[s T]HE MONSTER IS ALREADY CONNECTED!":RETURN
3530 PRINT"[s O].[s K]. [s I] CONNECT THE ELECTRODES TO THE MONSTER!"
3540 AR=2:FOR X=1 TO 5:IF V$(X)=G$(23) THEN V$(X)=""
3550 NEXT:RETURN
3560 PRINT"[CLEAR][DOWN][DOWN][DOWN][s T]HE MONSTER SINKS INTO THE PEAT BOG."
3570 PRINT"[DOWN][DOWN][DOWN][BLACK][s W]ELL [s D]ONE. [s Y]OU HAVE SOLVED THIS ADVENTURE."
```

BOUNCES
Beyond C-64 £9.95

Originality 10 · Playability 8 · Graphics 5 · Value For Money 8



THERE HAS BEEN A LARGE number of 'fighting' programs appearing over recent months, we've had kung-fu, wrestling and shooting. Now Beyond has brought out a game that combines the elements of a 'fighting' game with those that expect you to get a ball in the goal, as in football.

You find your player in an arena, you at one side and your opponent at the other. Both players are tied to the wall behind them by a length of elastic, and are armed with some sort of scoop with which you can try to catch the ball that is flying around the screen.

Points are awarded for getting the ball in the net, which is situated at the top of the screen or for knocking your opponent over.

What makes this game so much fun is watching your players floundering on the floor being dragged along by the elastic and seeing them being knocked senseless by the walls at either end.

There really isn't a lot more that can be said about this game apart from the fact that the idea is extremely original and that Beyond has created a game that is great fun to play.

S.C.

IWO JIMA
PSS £7.95 Joystick Required

Originality 7 · Playability 6 · Graphics 8 · Value For Money 7



ONE OF THE BLOODIEST battles in the Second World War was the attempt by the US Marines to regain the island of Iwo Jima from the Japanese. John Wayne managed it. Now, thanks to the title in the PSS Wargames Series, you too can see if you have a flair for tactics.

You have between 32 and 36 turns to eliminate all the Japanese forces from the island depending on which of the five skill levels you select. Tactics must be selected carefully as both all out attack and solid defence are likely to meet with failure. Whilst you receive re-inforcements (weather permitting), the Japanese don't and so tend to launch suicide attacks when a unit is close to being wiped out.

The game is controlled entirely via a joystick using it to select from your various units and determine their options from assorted menus. These options include landing on one of the six beaches, move, attack or pass. Each unit has aggression and defensive factors, a movement allowance and a combat range.

Iwo Jima is a very user friendly wargame and I found it more challenging to play than its sister game Falklands 82.

G.R.H.

KONAMI'S PING PONG
Imagine £9.95

Originality 8 · Playability 4 · Graphics 8 · Value For Money 7



IN THE DIM AND DISTANT past the only video game available was a crude table tennis simulation called Ping Pong. Do not confuse it with this little pearl from Imagine.

The opening screen is typical Japanese cuteness where a ping-pong ball bounces out Konami's name and then bounces down on to the head of an unsuspecting mole. After selecting the level of play the next bold graphic display shows the view of the table from the human player's end with an audience just off in the wings. At each end of the table is a disembodied hand grasping a bat.

During play the bats follow the path of the ball automatically and the only control that you need worry about is the kind of stroke to play.

Occasionally a stroke will result in a 'floater', a weak shot which suggests a superfast smash as a reply. In the lower levels this is no problem but as you progress through the levels, the game gets faster and faster and relies more on reflex action just like the real thing.

The only qualms about this excellent conversion is the lasting appeal of the game.

ED

SKYHAWK
Bug Byte £2.95 C16

Originality 3 · Playability 7 · Graphics 7 · Value For Money 9



DURING THE CURRENT burst of interest in the C-16/Plus/4 market a lot of C-64 games are gaining a new lease of life. Skyhawk resembles Virgin Games' Falcon Patrol which has long been a favourite of mine.

As the pilot of an advanced VTOL fighter it is your job to see off the enemy bombers which are devastating the countryside. Fuel and air-to-air missiles (AAMs) must be monitored carefully during each battle and should the need for re-equipping your fighter arise you will have to be fast and careful and keep an eye on the radar for approaching bombers.

At first each wave consists of two planes but this gradually increases to four as the game progresses.

Unlike Falcon Patrol, the jet cannot descend into the 3D scrolling landscape. This makes refuelling less hazardous, but still tricky, because the plane cannot crash unless it collides with a bomber.

Skyhawk is a game of survival. The bombers keep coming until you have run out of jets or fuel after all of the refuelling platforms have been destroyed. Cheap, cheerful and quite addictive.

E.D.

BANDITS AT ZERO
Mad £2.99 C-16 + Joystick

Originality 3 · Playability 4 · Graphics 4 · Value For Money 4



AIR ATTACKS FROM AIRcraft carriers can best be stopped by sinking the floating airstrip which harbours the planes. Sound logic but not as easy as it seems. First you have to battle through the waves of fighters which protect the ship. This is your mission in Bandits at Zero and it is quite a challenge.

The game starts with your solo fighter-bomber flying low over the sea searching for the enemy. First there is an encounter with a refuelling plane to top up your tanks ready for the long day ahead but soon your radar display becomes live with tiny blips which denote the enemy fighters ahead. As the plane flies to join combat the screen scrolls smoothly as a coatline seascape is gradually unfurled, but this is no tourist trip because soon the enemy is there in front of you.

The first day is easy pickings because most of the planes are flying in the same direction as yourself but more slowly. Taken by surprise, the pilots rarely fire back and the only real problem is your own skill at avoiding collisions with them. A cautious eye should be kept on the radar screen for the occasional attacking plane which zeros in on you from the opposite direction at high speed. If a collision occurs you lose one of your seven protective shields.

After this leisurely cruise across the rolling seascape, night gradually falls (nice use of colour as the daylight gradually fades). By now your fuel is critically low but there is help at hand as you rendezvous with the tanker once more. Tricky business, this aerial refuelling lark. Altitude and speed are critical as the umbilical cord is attached and if you take too long dawn will break and the tanker will zoom off to safety. Enemy attacks are few at night but occasionally a fighter will appear. The best policy is prayer. Fuel is more important than fighting so with a little bit of Divine intervention and a gritting of your teeth as you bite the cord, the night will pass without the loss of another shield and the sun will come up on a freshly refuelled jet eager for another day's action.

The new day brings with it a new breed of pilot, the rear guns blaze in your face and their rockets spit out at you if they get behind your plane. More jets appear to be flying towards you at high velocity and your shield count is in danger. Night seems a long way off as you fly on towards your target across the never-ending sea.

Another nocturnal refuelling finds you closer still to your target. This is indicated by the fleet of battleships which are sending up a battery of shells. Still the enemy planes attack and the tiny black flecks of flak threaten to blot out the sun.

Before long you begin to wish that day three had not dawned and regret that it did.

Graphically, the game appears a lot more sophisticated than it actually is and in comparison to many other C-16 games it shows what a thoughtful programmer can do within the cramped memory confines of the machine. Adding an extra problem to deal with as each screen goes by, helps to stave off the boredom of a straight forward shoot-em-up.

The only really weak point of the game is the music which is mournful and unimaginative. It only appears at the end of each day so I won't complain too loudly.

Ammunition supplies are reminiscent of a cowboy's six-gun in the old silent movie days, bullets are unlimited. In this way the temptation to fly with trigger finger down would have to be curbed in favour of accurate and thoughtful gunnery skills.

Don't allow these criticisms to draw you away from the fact that this is a superb game worth every penny of your hard earned cash. If things do get a little dull you can sweep your jet back and forth across the skies.

In this way the program commits more enemy planes into flying towards you at a high rate of knots increasing your problems three-fold.

J.G.

TIGERS IN THE SNOW

US Gold £9.95

Originality 5 Playability 8 Graphics 5 Value For Money 7



IN DECEMBER 1944 HITLER'S army was being driven back towards the Fatherland and a major counter offensive was needed to drive the allies away from the Rhine. The conflict that ensued became known as the Battle of the Bulge and marked the beginning of the end of World War II.

A major part in the offensive was played by the Panzer Tank Divisions of the German army. These powerful tanks had earned themselves the nickname of 'tigers', hence the title of this simulation.

As in the real battle the action begins on 16 December and ends on 27 December. Players can take charge of each army or the computer will play the role of the enemy.

The weather conditions of those fateful days of long ago are faithfully reproduced during the game, determining whether your airborne supplies can be flown in or not. Being Northern Europe in the dead of winter, you soon learn to take advantage of the few clear days that occur. Supply level affects the fighting strength of your units and can become a crucial factor in your success.

The situation at the beginning of the game is that the Allied Forces are represented by greenhorn American troops against a very experienced and strong German army. The initial aim of the Allies is to hold back the onslaught until reinforcements arrive. For the Germans it is to break through and cross the River Meuse, taking the major towns of Bastogne, Marche and Rochefort on the way for maximum points. The eventual aim of the Allies is to breakthrough to the East.

Each day is broken down into a sequence of five moves. Firstly the Germans alter their positions and enter into such skirmishes as these moves allow. Next it is the Allies turn to do the same. After this, reinforcements are placed on the map and Victory Status is assessed to see if the game can continue. Finally, you are given the option to save the game before the next day dawns.

The result of a battle is determined principally on the relative strength of the units involved and the strategy chosen for attach and defence. If the result of a battle is a dramatic win, the victorious unit may advance three squares and attack again later that day. Defeated forces may retreat if there are any combat points left. Sometimes this will overtax an undersupplied unit and they will be wiped out.

Attack strategy can be a major offensive, a medium battle, a light skirmish or a gnat bite of a reconnaissance mission. In defence you may chose to counter-attack, try to hold your position, withdraw or try delaying tactics. Choosing the correct response to your opponents strategy is the aim so that you will lose fewer combat points, if any, and live to fight another day. The longer a unit survives the greater the number of combat ponts that are awarded at the end of each day. A unit with fewer than 25 points is in serious trouble and must rely on plenty of movement points to pull back out of harm's way while they recover. This I learned from bitter experience!

Extra combat points can be awarded if you chose to include an artillery barrage with your attack. Each day you have a number of artillery points awarded and you are given the option of using a limited amount of fire power to shore up a flagging unit.

The documentation which accompanies the game is extensive but confusing. A map of the whole area is also supplied but I found it to be confusing. Locating my starting position took a while and this wasn't helped by the key being some distance from the map itself.

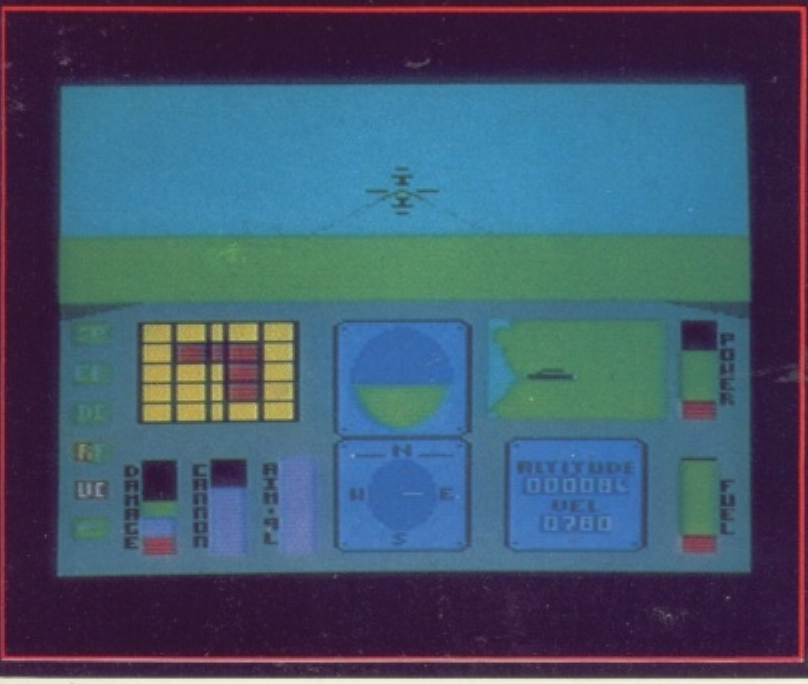
Eventually, I struggled through the documentation and found that the best way to learn is through experience. I fought long and hard but I regret to say that if it had been left up to me we'd all be singing German folk songs by now! This is an enjoyable simulation but this battle is popular for computer strategy games and I have seen better in my time.

E.D.

WING COMMANDER

Mastertronic £1.99

Originality 3 Playability 4 Graphics 6 Value For Money 6



YES, IT'S YET ANOTHER flight simulator! This one sees you in charge of a jet fighter defending your island from the enemy planes. Their target is the nuclear power plant and it is vital that you intercept their bombers before they reach their destination.

The screen display is in two halves, the top window showing the view from your cockpit whilst the bottom part of the screen contains your instruments. These include a radar, compass, several guages and warning indicators and a scrolling map of part of the island. A full screen map of the entire island can be called up at the press of a button.

Although the instruments look complicated, controlling your plane is very simple. You can attack an enemy plane with either cannon or missile, but you have finite supplies of both. You also get the chance to refuel in mid-air providing that certain conditions are met.

Wing Commander was written back in 1984 and is beginning to look a bit dated now but if you want to try a cheap flight simulator before splashing out on a more sophisticated model, it could be just the thing you are looking for.

G.H.

OFF THE HOOK

Electric Dreams £6.99 C64



PRODUCING COMPILation tapes for charity organisations is becoming popular. Off The Hook is yet another and all proceeds from the tape are going to the Prinnce's Trust for the rehabilitation of drug addicts.

It is very difficult to comment on a tape that is being sold for charity as you obviously want the people involved to sell as many tapes as possible. Thankfully in the case of Off The Hook the 10 programs included are all of a fairly high quality.

Programs included on the tape are the second of Activisions programs featuring Pitfall Harry and is great fun to play. Harry is trying to rescue his niece – makes a change from Princess – while avoiding the many dangers on his way such as scorpions, giant frogs and electric eels. This game kept me occupied for a long time, never dies may have something to do with this!

Next on the cassette is Space Pilot 2 from Activision. This game offers nothing out of the ordinary and looks a little dated. Nevertheless flying your space ship over a scrolling backdrop and shooting enemy ships is still great fun.

Probably the largest and most complex game on this cassette is Psytron from Beyond. When this game was launched on the Spectrum around two years ago I didn't get anywhere with it and I've not improved in those two years. You are in charge of the Betula 5 installation and must defend it against enemy invaders. This means that you must shoot them out of the skies, go after any that get into your buildings and see to any repairs that your complex needs. Psytron is a very complex game – the fact that the instructions take up more space than all of the other games on the cassette put together shows this.

Fall Guy from Elite allows you to play that famous TV character Colt Seavers the stunt man. Your job is to perform all the stunts set in as little time as possible. Nothing really exciting here but well worth loading in for a quick half hour.

Demons of Topaz from Firebird places you on the asteroid Topaz. Your aim is to find the sacred crystals that are scattered around the many levels of the asteroid. Yes you've guessed it, Demons of Topaz is a platform game. It is however quite well produced and is good fun to play.

Sheep in Space from Llamasoft should need no introduction. But, just in case you've never heard of it, it's a little similar to a Defender game, but in this case there's no one to rescue and you must kill all of the Hostiles before they can blow up your planet. Oh by the way, you aren't given a spaceship, this time you are an intergalactic sheep.

Kong Strikes Back from Ocean is probably the most disappointing game on the cassette. Mind you, when you think of the price it's not all that bad. In this game you must guide your man up the fairground track to rescue your damsel in distress. Of course there are objects for you to avoid on your way up such as roller coaster cars and springs.

Black Thunder from Quicksilva finds you charging along a number of roads shooting evrything in sight. An extremely fast game and great fun to play.

Death Star Interceptor from System 3 is the penultimate game on the cassette. This game finds you once again in control of a spaceship trying to save the galaxy. Again nothing out of the ordinary is offered but if you are into zapping aliens then you'll probably enjoy it.

Bringing up the rear is Talledega, and that's exactly what I did in this racing game. and this one offers nothing out of the ordinary though it is great fun to play.

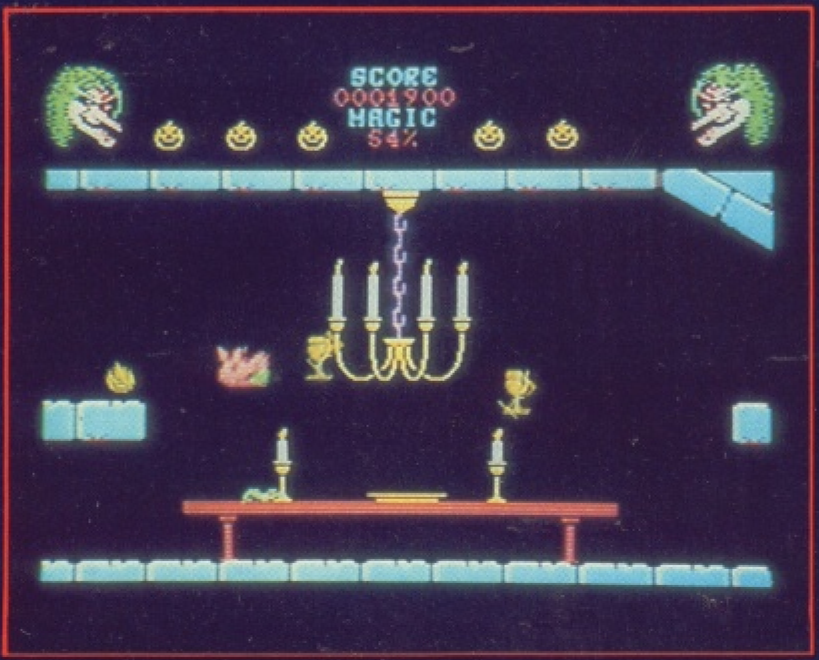
When you consider the amount of money that is being asked for this cassette and then work out how much is being asked for each game it's impossible to complain.

Even if there are only a couple of games on the cassette that you haven't already got then its still worth the asking price. Go out and buy it NOW. **S.C.**

CAULDRON II

Palace Software £7.95 Joystick required

Originality 8 Playability 7 Graphics 8 Value For Money 7



ANYONE WHO DEFEATED the evil pumpkin in the original Cauldron now has the chance to turn the tables. The Witch Queen is in charge of a wholly evil regime. The only person who can put things right again is you – a brave pumpkin warrior!

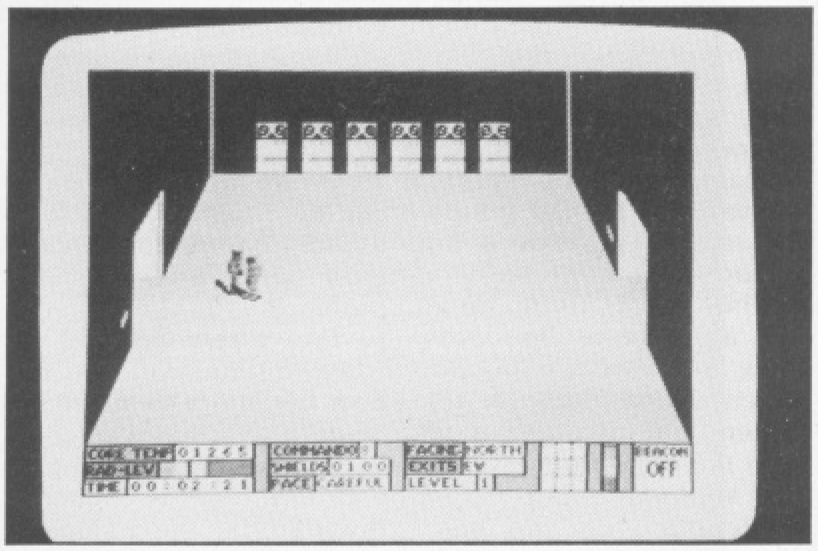
Cauldron II is a sort of platform game but instead of jumping, you bounce and this is likely to take you some time to get used to as our hero rebounds off walls like a ball bearing on a pin table. Starting in one of several randomly chosen locations, you must collect assorted objects before you can depose the queen. These include crown, scissors and axe. There are various nasties trying to stop you and colliding with these depletes your energy. You can however pick up glowing spheres which both replenishes your energy and allows you to fight back by hurling these spheres at your enemies.

The graphics are good and amusing with some lovely touches such as when a gargoyle tips you off a platform into open space. But I would have preferred it if everything scrolled instead of jumping from screen to screen – you are never quite sure what nasties lie in wait for you. **G.H.**

COUNTDOWN TO MELTDOWN

Mastertronic MAD Range £2.99 Joystick required

Originality 7 · Playability 6 · Graphics 3 · Value For Money 7



AFTER AN EXPLOSION IN A nuclear power plant, the central core is overheating and there is a considerable danger of a second blast. Your task is to dampen down the reactor rods using a team of remote controlled android commandos.

You have eight robots to manipulate, each with a different set of skills and must guide them through the 2000 rooms on eight levels as you try to reach the core in time. There are assorted intruders that must be disposed of before they sap your strength and mobility etc and render that particular robot useless until he can be repaired by one of his companions.

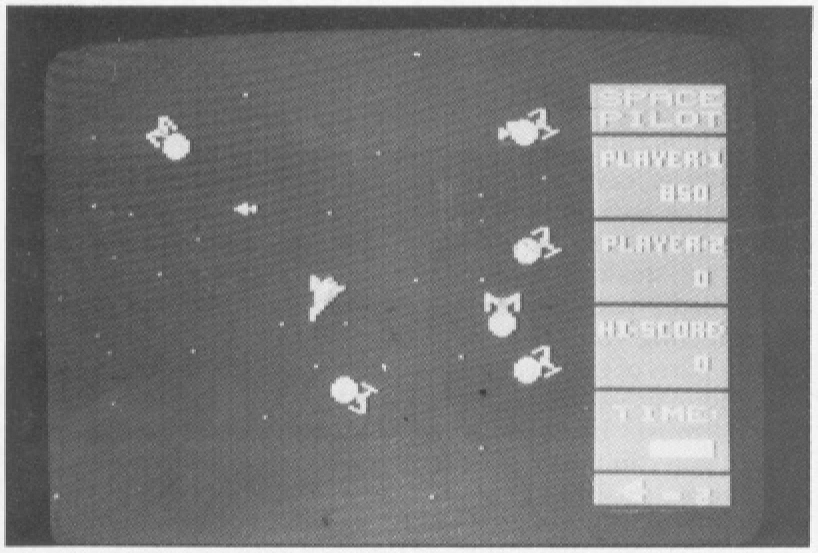
Moving the robots and firing is via the joystick but there are also several keyboard commands for using the objects that you have found. The graphics are fairly simplistic – 3D views of the rooms with exits and nasties shown but you can switch between views to show the doors more clearly.

Countdown is an interesting game that will keep you quiet for ages – you can save your current position. If you enjoy large scale mapping games, this is good value for money. **G.H.**

SPACE PILOT

Anco/Kingsoft C-16 + Joystick £5.95

Originality 3 · Playability 2 · Graphics 5 · Value For Money 3



SPACE PILOT STYLE GAMES have been with us for many years but this is the first that I have seen for the C-16.

The game is one of survival as you roam the galaxy facing the attacks of innumerable alien craft. Flying singly or in formation these demons are armed with heat-seeking missiles which will attemp to snuggle up close to your ship, vapourising you as they do so.

The background of stars scroll with a beautiful 3D effect in every conceivable direction but this is where the price of sophistication must be paid. All this activity results in a deadening effect on the spaceship's movement lessening the excitement which the gameply has to offer.

It is also noticeable that the complexity of the angled scroll is slower than the simpler vertical or horizontal scroll but I will accept this as a game 'feature'.

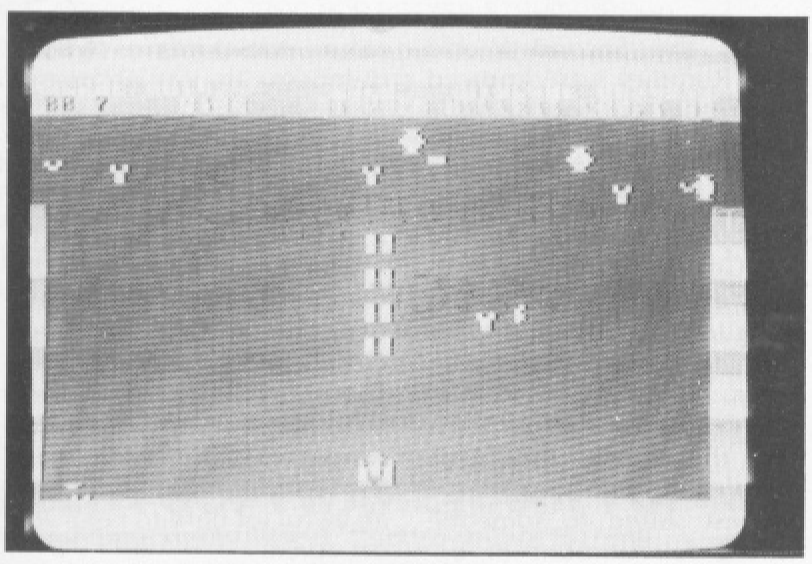
I do feel that the game is playable and certainly stunning. Each of the four waves of aliens are armed with missiles which behave in their own characteristic way.

To use a cliche, this is a flawed masterpiece. **E.D.**

SOLO

Bug-Byte C-16 + Joystick £2.95

Originality 2 · Playability 3 · Graphics 2 · Value For Money 3



PROGRAMMER STEVEN Kellett's name keeps cropping up on C-16 games. He specialises in unsophiscticated shoot-em-ups which rely purely on fast reflexes for survival.

This time he appears on the Bug-Byte label with a typical product of his endeavours which provides mayhem with a storyline.

Solo is the fighter you control in your battle against the aliens who appear in droves to annihilate you. The action is swift as they hurl everything at you in their venomous attack. Solo can move in any direction on the screen.

At the beginning your craft sits on the bottom border but you'd be foolhardy to sit there too long. In a matter of seconds the screen explodes into frenzied activity and a second wave of attack will start before you have wiped out the first.

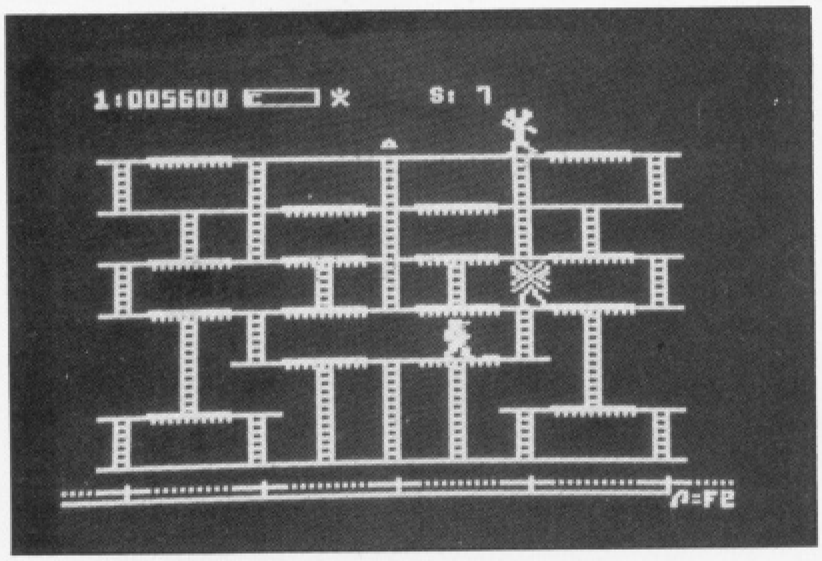
There are 10 waves of terror to test your staying power and I suspect that survival through all of them is impossible unless you have nerves of steel and reflexes honed to superhuman efficiency.

Lovers of unpretentious zapping games will love this. **E.D.**

THE CHIP FACTORY

Supersoft C-16 + Joystick £6.95

Originality 2 · Playability 3 · Graphics 2 · Value For Money 3



CHARLIE IS A TRAINEE accountant who sneaks into the chip manufacturing plant to try his skill at production. Bugs, sparks and the odd spanner in the works run rampant at night and it is up to you to guide the lad around the screen to produce his microchips.

This is a platform game and a conveyor belt runs along the bottom of the screen. On the conveyor are sockets and Charlie has to drop the chips from level to level so that they fall into an empty place on the belt. When all of the places on the belt are filled he must climb to the switch at the top of the screen to wind the belt on.

The nasties also wander around the platforms and contact with them can be fatal.

Completely filling a conveyor belt allows Charlie to move on to a new production line and further problems.

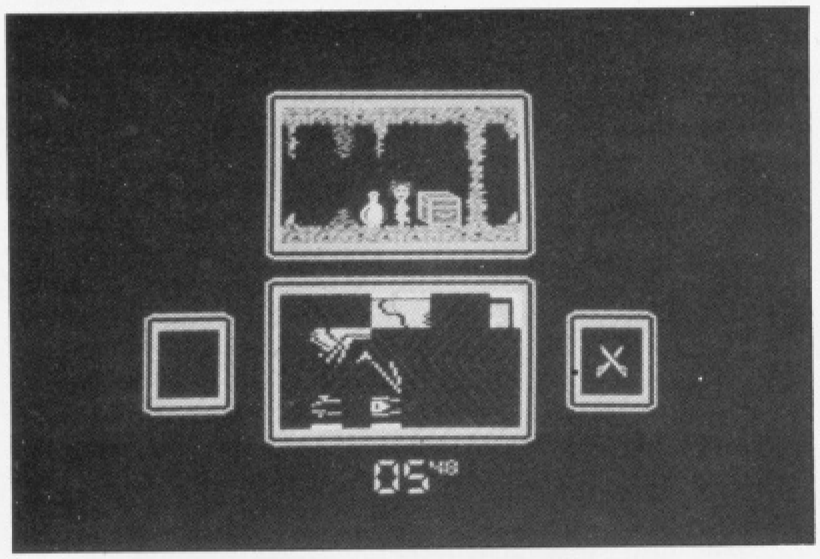
The graphics could be more imaginative but this does not affect the game too much. The action is certainly frantic and the planning of your next move is frequently thwarted by the numerous nasties. Simple but effective.

J.G.

HOCUS FOCUS

Quicksilva £8.95 Joystick optional C64

Originality 4 · Playability 3 · Graphics 5 · Value For Money 5



JEREMY IS A YOP PHOTOgrapher working for the Daily Shocker and is given the assignment of taking pictures of the Potty Professor's inventions in his hidden lab. The whole cave system is radioactive and protected by genetic mutants who are intent on stealing your film and sword – your one means of protecting yourself.

The parts of the inventions are hidden in assorted objects which must be searched before you can take a picture of them. Some objects contain ghostlike figures that again steal your film. When you have some pictures ready to develop, you can make your way back to the start where you get the chance to place the pictures on to a large grid.

Moving Jeremy is a simple left, right and jump although there are several other functions which are icon driven. These are: examine an object, take a photo, pick up or switch on an object and fight.

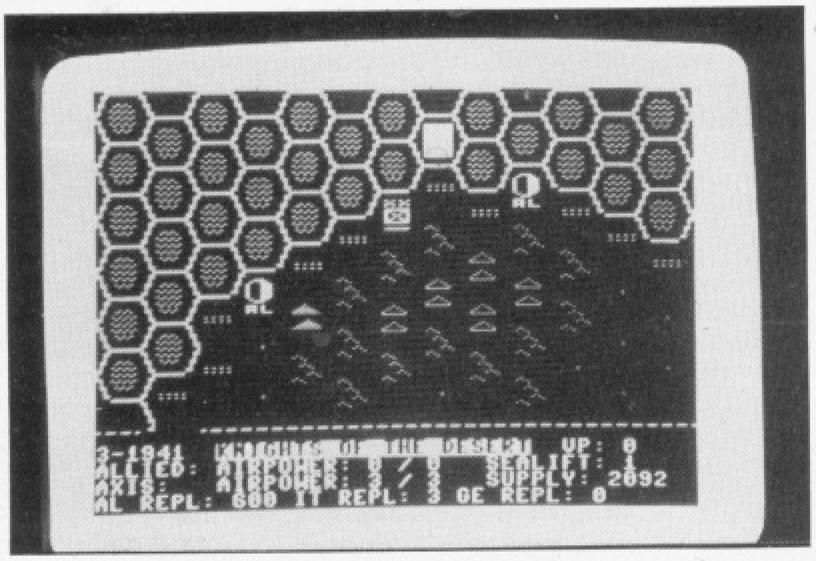
Icons are supposed to make things easier to manipulate, but they just don't work here.

G.H.

KNIGHTS OF THE DESERT

US Gold/SSI £9.95 cassette, £14.95 disk C64

Originality 6 · Playability 5 · Graphics 4 · Value For Money 5



DESPITE THE FACT THAT HE was on the "other side" during the Second World War, Rommel was renowed and respected for being a decent fellow and a brilliant commander. It was only the fact that the British, under Montgomery, had cracked the German codes and so knew every move that he was going to make, that led to his ultimate defeat in North Africa.

The first thing to be said is that this is an incredibly complex wargame and would be best suited to someone with a fair amount of experience or a lot of patience. Players take it in turn to reinforce, supply, move and fight as the Germans attempt to take Alexandria whilst defending their main base of El Agheila.

The game is played on a hex grid that whilst not exactly graphically stunning, is perfectly adequate. The instructions are long and complex (a book would have been better than the huge closely printed sheet) and the gameplay is hardly user-friendly. But, if you enjoy wargames or are interested in the historical aspects, there is an awful lot here to keep you occupied.

G.H.

Your COMMODORE

YOUR BEST INDEPENDENT COMMODORE MAGAZINE

SO YOU OWN A COMMODORE?

SO YOU'VE WRITTEN SOME PROGRAMS?

SO WHY HAVEN'T YOU SUBMITTED THEM TO US?

YOUR COMMODORE IS ALWAYS ON the lookout for new material for publication and we know that there are thousands of intelligent, literate, innovative and creative Commodore owners out there, so why don't we get together?

If you have written a program, be it a game or a utility, that you think you should share with other Commodore owners, then why not submit it to us for publication. Don't forget we pay very well for any articles that we publish.

All submissions should be well documented and where possible include a line by line breakdown of the program together with a list of any variables used. A copy of the program should be included on either tape or disk.

All submissions should be sent to the address below. If it is not chosen for publications then it will be returned to you.

You may not have written any software yourself, but you may have very firm opinions about the world of Commodore and all its attendant industries and products. If you do, then put your views or questions on paper and post them to us again at the address below – you might even get paid for airing your views!

All submissions should be sent to:

The Editor
Your Commodore
Argus Specialist Publications
No 1 Golden Square
London W1R 3AB

Office Use Only

Surname: ....

Firstnames: ....

Address: ....

....

Postcode: .... **Telephone Number:** ....

Times to contact you: ....

Program Name: ....

Computer/memory size: ....

Extras required (disk etc): ....

Have you sent the program to another magazine? Yes/No

PLEASE COMPLETE IN BLOCK CAPITALS

Date received: .... **Date ack. sent:** ....

Date checked: .... **Checked by:** ....

Rating: ....

Comments: ....

....

Reply sent: .... **Ack. received:** ....

Date printed: .... **Date returned:** ....

Fee: ....

Date Fee sent: ....

INCENTIVE



THE GRAPHIC ADVENTURE CREATOR

The amazing, combined text and graphics adventure generator enables you to produce fast, compact adventure games with the minimum of effort and the maximum of creativity. Stunning graphics can be created quickly and easily with the powerful picture generator that has many features including:– dot, circle, elastic line, ellipse, fast fill, shading, step by step review, easy editing and the ability to merge pictures.

The comprehensive, intelligent command interpreter can handle complex sentences and multiple input commands. There is also an extensive text compression facility that allows you to produce far more detailed and numerous location descriptions. Plus a full function editor, automatic word formatting, a logical command interpreter and an abbreviated input acceptance facility.

So unleash the power of your imagination now with **The Graphic Adventure Creator,** for your Commodore 64 computer. Available from all leading retailers or direct from Incentive on 0734 591678.

INCENTIVE SOFTWARE LTD. 54 London Street, Reading RG1 4SQ.

Please rush me, for my Commodore 64 Computer **THE GRAPHIC ADVENTURE CREATOR**

☐ Cassette £22.95
☐ Disc £27.95

BARCLAYCARD VISA

I enclose Cheque/PO for £________

or please debit my Credit Card

No.________

Name________

Address________

George Duval has managed to get hold of some Amiga games software. Read on to find out what's available.

IT IS A REAL SHAME THAT COMMODORE, in all its wisdom, has decided that the Amiga should be a business machine. When you think about it, with amazing graphics, 4096 colours, stereo sound and a 68000 processor, there cannot be a machine more suited to high quality games.

Fortunately, many of the large American software houses agree with me, and slowly but surely some excellent products are appearing. Electronic Arts was the first company to produce games for the Amiga, not surprisingly perhaps – Commodore gave development machines to Electronic Arts more than six months before anyone else!

Most 64 owners will recognise the first three games EA released – Archon, One-on-One, and Seven Cities Of Gold as they are all conversions from C64 originals. None of these products take full advantage of the Amiga's capabilities, but each has its own touches that make it just that little bit better than anything available for a normal home micro. But then the Amiga is no 'normal home micro'!

Seven Cities of Gold was the first complete game for the Amiga, however it is also the least good. It is ridiculously similar to the 64 version, even down to the chunky horizontal scrolling; which is more reminiscent of the Vic 20 than a £1500 Amiga!

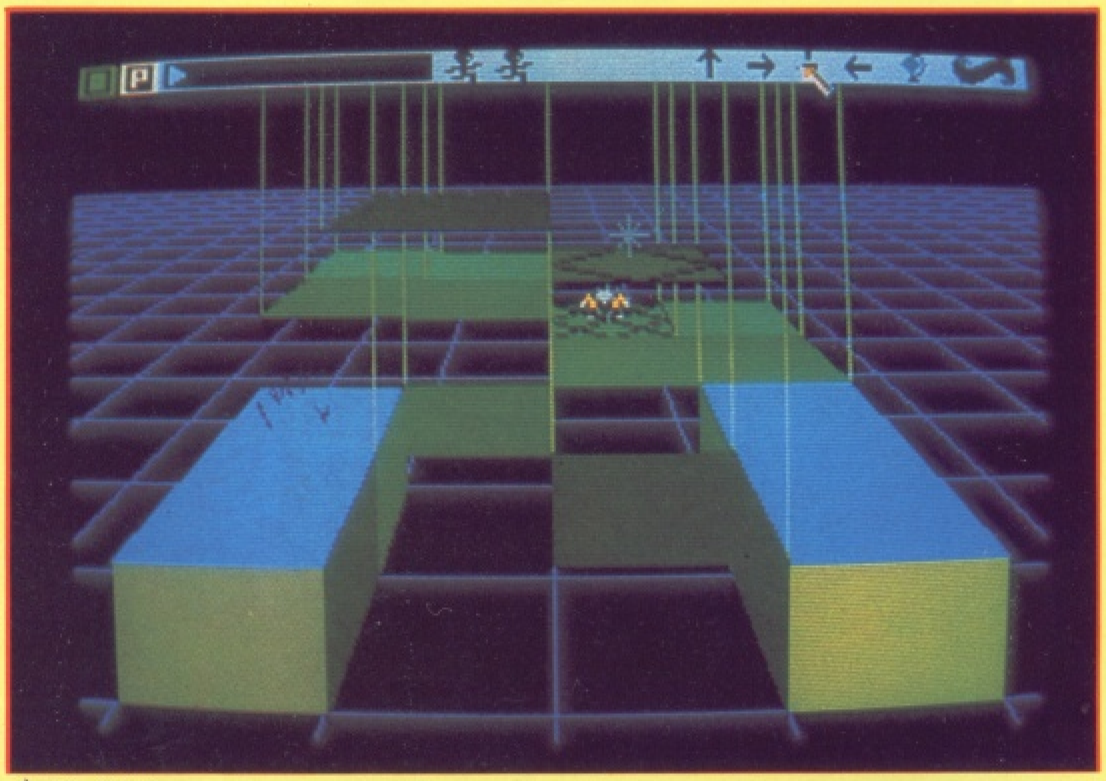
In The Basket

Things improve greatly, however, with One-on-One. Although it too is a straight conversion from a well known 64 game, the graphics are good and the sound is absolutely mindblowing! In case you don't know the game, it's a basketball simulation, except that you don't play a whole team, just one player – Dr J, or Larry Bird (both of whom I am assured are well known basketball stars!). You must try and out-fox your opponent and score as many points as possible in the time allotted.

Graphically One-on-One is good, though by no means special. By using vertical sprites, the programmers have taken the easy way out. Had they used the much vaunted 'Blitter chip' the end result would have been truly outstanding. The sound however is a different story. Using sound-sampling EA has managed to use sounds from a real game, and everything from the ball's bounce to the popcorn seller is fabulous.



▲ Arctic Fox



▲ Wynd Wälker

Adventuring

Perhaps the best known of EA's conversions is Archon. As a 64 game I rated it very highly, since it managed to combine the strategy of chess together with a more exciting 'arcade' section. On the Amiga it is the same game only better. The graphics are beautifully defined, the sound is fun, and the gameplay is gripping – what more could you ask for?

Out In The Cold

Arcticfox! If the first three games are good conversions, then Articfox is what the Amiga is all about. Programmed by Dynamix, it must be the best game on any personal computer. To describe the game in full would take hours, but briefly it is true 3-D (as opposed to sprite 3-D) Battlezone type game, in which you control a 37 ton missile launching 100 mph tank! This is the only game I have seen that shows what can be done on the Amiga. It has amazing graphics, great sound, and is immensely playable.

Flight Simulator ▼



AMIGA



PLAYTIME

On the packaging of Arcticfox is a line which I think sums up what this game is all about. It reads "Where do you sleep when you own a 37-ton tank? Anywhere you want to"!

Arcticfox's mission is set in 2005, and you must infiltrate the Alien's force field and blow up the main fort, thereby defeating the aliens. Against you are a wide variety of stationary as well as mobile weapons which will do their utmost to make sure you get massacred well before you reach the main fort.

What makes Arcticfox so good is the attention to detail. The instrument panel alone is amazing, with each of the weapons you have being activated by moving an on-screen hand. Radar too has been implemented, and the mini screen used for displaying the guided missiles is incredible.

Reviewing this game poses one difficulty, what superlatives are there left to use? Screen shots cannot do this game justice, and even the fact that it is quite slow doesn't alter the fact that this game is great.

In On The Act

Although EA is the biggest producer of Amiga software, other American software houses saw this computer as their chance to be 'there at the beginning'. Activision managed to release three titles very quickly, yet retained a very high standard.



▲ Borrowed Time

As with most of EA's games, Activision has converted three best selling Commodore 64 titles; Hacker, Borrowed Time, and Mindshadow.

In 1985, Hacker was one of Activision's best selling games. On the Amiga, it has been upgraded graphically, but the gameplay remains the same. You have broken into a computer and must travel around the world collecting sections of a secret document. What makes this game so good on the Amiga is that the small 'monitor' within the game, which is used to display the locations, shows some incredibly well defined pictures of all the major cities around the world (Tower bridge is especially good!).

Mindshadow and Borrowed Time are both graphical adventures, and although they have no sound, the graphics are of a very high standard. In Borrowed Time you play a 1930s detective called Sam Harlowe, and it is your job to crack a complicated case, and avoid being murdered – not an easy task!

Mindshadow is a more impressive adventure, in which you play a victim of amnesia who must find out who and where he is. Although neither of these games use the full potential of the Amiga, they were created fast, and as such bode well for what Activision will do in the future.

Although England is well behind on Amiga development, we do have one game – Brataccus from Liverpool's Psygnosis. Originally a QL game, more recently it has been released for the Atari 520ST, Mac and Amiga. It is an odd game, in which you must wander around a spacecraft, doing battle using your sword and refusing offers of drinks at the bar! Unfortunately Brataccus is another example of a game that does not take advantage of ANY of the Amiga's facilities, even down to the reduced screen size and limited colour.

◀ Archon

Hacker ▼



What Next?

What the future holds for Amiga games is uncertain, but I have seen sneak previews of two games which should be released towards the end of 1986. Wynd Walker will be Commodore's first, and quite possibly only, game for the Amiga. It is an arcade adventure with you playing the hero, a wizard. As yet Wynd Walker has no gameplay, but enormous potential.

The Amiga is most suited to 3-D simulations, and I have seen an unfinished flight simulator that, when it is released, will have airline pilots aghast! Programmed by the team responsible for the now legendary Flight Simulator II for the IBM PC, it is fast, has wonderful solid objects and great potential. Unfortunately in the version I have, you can also fly UNDER the runway, and THROUGH the Pyramids!

As yet, no one has written a game purely for the Amiga, using all the potential of this incredible machine. However I know for a fact that EA is writing the arcade class Marble Madness, which should be out within two to three months. All eyes are on it and another EA title Return To Atlantis – the first game to make use of the Blitter – to show what really can be done.

Get in a spin with W Bremner's insight into your 1541 disk drive.

PROGRAMMING THE 1541

IN ORDER TO INCREASE your understanding of the 1541 it is necessary to comprehend how the user communicates with the 1541. This article should give you a better grasp of this subject.

The program listed at the end will be used throughout the series to investigate the drive, and allows the user to store and retrieve programs and data within the drive's own memory. The program is rather slow, as it is written in Basic (with some machine code), but is fully documented, and serves to teach as well as being a useful tool. Readers interested in acquiring more complex utilities should get in touch with Evesham Micros on 0386 41989 or obtain a copy of Drivemon by Starpoint Software.

The Serial Bus

The 1541 Disk Drive is one of the many peripherals available for the Commodore 64 which use the serial bus to send and receive data to or from the host computer. The bus is a daisy chain arrangement allowing up to five peripheral devices to be connected together, for example, one printer and two disk drives. Since several devices may be connected at the same time, there must be a way of distinguishing one device from another. This is the purpose of the device number, and the 64 assigns numbers zero to 31 to the serial bus, eight to 11 covering up to four drives.

The 64, known as the bus controller, can command a device to receive data on the bus (LISTEN) or transmit data (TALK). Only one device may talk on the bus at a time, but many devices may listen to one talker, and the 64 is the only device on the bus that may act as controller. When a device is addressed, the 64 sends an attention signal (ATN) over the bus, alerting all connected devices that they should be aware that data communication is being established with one of them. The 64 then sends the number of the desired device, and if that device is present, it will respond to the ATN, otherwise a time out error will occur, and the 64 will report a DEVICE NOT PRESENT.

The 64 then indicates to the selected device whether it should be a listener, and receive data from the bus, or be a talker, and send data. A second address may also be sent to perform any further set up operations. To ensure the accuracy of the following data transmission, the data is sent over the bus one character at a time. Only when the receiver acknowledges the accurate receipt of the data may the sender transmit another character. This fairly lengthy procedure is known as handshaking, and is necessary due to the 64 and 1541 not being centrally synchronised. When the data transmission is complete, the device is de-addressed; if the device was sending data, the 64 sends an UNTALK command; if the device was receiving data, the 64 sends an UNLISTEN command. The bus is then free to handle the next transmission.

Controlling the 1541

Various housekeeping duties such as renaming a file can be executed by sending special commands to the 1541. In order for us to manipulate the drive's memory, the Disk Operating System (DOS) designers kindly included some extra commands, some vital, some seemingly superfluous. The three most important of these allow us to store, retrieve and execute machine code routines and data within the 1541 RAM. They are similar to the POKE, PEEK and SYS functions in Commodore Basic, and work in much the same way. These special commands along with their relevant parameters are sent to the drive along the command channel (15), and are covered in depth in the 1541 user manual, so I will only briefly review their syntax:

MEMORY-READ; fetches up to 255 bytes from anywhere in the drive's memory, and returns them along the command channel.
Format: "M-R";CHR$(LO Addr.);CHR$(HI Addr.);CHR $(No.bytes)-optional.
MEMORY-WRITE: stores up to 34 bytes at a time in RAM.
Format: "M-W";CHR$(LO); CHR$(HI);CHR$(No.bytes); CHR$(data).
MEMORY-EXECUTE: executes a ROM or RAM routine within the drive's memory.
Format: "M-E";CHR$(LO); CHR$(HI).

Drive RAM Usage

$000 — Zero Page
$0100 — Processor Stack area
$0200 — Serial Bus Input/Output buffers
$0300 — Buffer#0
$0400 — Buffer#1
$0500 — Buffer#2
$0600 — Buffer#3
$0700 — Buffer#4-Used by DOS for BAM

The following program demonstrates the use of all the "MEMORY" commands in one sequence. A small routine is poked into Buffer #0 at $0300, which, once executed, stores a further byte in location 0400. Note that the m/c routine ends with an RTS (Return To Subroutine).

```
100 OPEN 2,8,15
110 PRINT#2, "M-W";CHR$
(00);CHR$(03);CHR$(6);CHR
$(169);CHR$(255); CHR$ (141)
;CHR$(00)CHR$
(04);CHR$(96)
120 PRINT # 2, "M-E";CHR$
(00);CHR$(03)
130 PRINT # 2, "M-R";CHR$
(00);CHR$(04);CHR$(01)
140 GET#2,B$:PRINT B$
```

The same routine when written in machine code is considerably more involved, but as long as the rules are strictly followed, in the correct sequence, equal results can be achieved, and often a lot faster. Luckily, most of the hard work has been done for us, and we only need to call the correct Kernal routines:

Serial Bus KERNAL Routines

LISTEN $FFB1 — Command a device on the serial bus to LISTEN.
SECOND $FF93 — Send secondary address after LISTEN.
CIOUT $FFA8 — Output a byte to the serial bus.
UNLSN $FFAE — Command all devices on the serial bus to UNLISTEN.
TALK $FFB4 — Command a device on the serial bus to TALK.
TKSA $FF96 — Send secondary address after TALK.
ACPTR $FFA5 — Input a byte from the serial bus.
UNTLK $FFAB — Command all devices on the serial bus to UNTALK.

Complimentary Bus Routines

IONIT $FF84 — Initialise Input/Output
READST $FFB7 — Read status word

To transmit data to a device, the accumulator (acc.) is loaded with the device number, and the LISTEN routine is called. The secondary address (channel number) is then stored in the acc., bits five and six are set (result = s.a. + $60) and the routine SECOND is called. Data characters stored in the acc. are then sent over the bus using CIOUT, and the whole sequence is terminated with the UNLSN routine, which sends an EOI (End Or Identify).

Getting a device to send data over the bus is just as easy. The KERNAL talk routines are used in place of their corresponding LISTEN calls, and data is input using ACPTR. Bit six of the status flag (updated after a READST call) can be checked after each ACPTR call to test for an EOI from the drive.

To open a file or direct-access buffer, bits five, six, seven and eight of the secondary address value are set (result = s.a. + $FO). Bits six, seven and eight are set (result = s.a. + $EO) to close the file/buffer.

Armed with this information, we can now write our machine code routine. Note that because the LISTEN/SECOND sequence is used three times I have set it aside as a complete routine call. In a program full of disk access, this is often a valuable space saving technique.

This covers all the "Memory" commands and their various peculiarities. As reference only, I have included details of those less well known commands I mentioned earlier. So far I have only encountered these commands implemented in elaborate disk protection schemes; they tend to suggest exact usage because of their limiting qualities. As in the case of the memory commands, they are executed by sending the function and its parameters along the command channel.

BLOCK-EXECUTE: Reads a specific track and sector into a previously opened buffer and executes the machine code routine at the start of that buffer.

Format: "B-E";CHR$(channel);CHR$(drive no.) ;CHR$(track);CHR$(sector)
E.g. PRINT #15, "B-E";2;0;18;10

USER-JUMP :Apart from the two "Ux" commands used to read (U1/UA) and write (U2/UB) specific sectors to and from the diskette, and the two "Ux" commands used to set the drive bus speed (U9/UI) and reset the drive (U10/UJ), there are six "user-jump" functions which, when called, execute code at the beginning of buffer # 2 ($0500).

U3(UC) — jump to $0500
U4(UD) — jump to $0503
U5(UE) — jump to $0506
U6(UF) — jump to $0509
U7(UG) — jump to $050C
U8(UH) — jump to $050F

The nature of these commands would suggest a "jump-table" set up to perform varied tasks within the 1541's memory, easily called up from an external Basic or machine code routine. For example, a routine which reads section 18,0 into buffer#1 and sends it at high speed along the bus could possibly be called by:

PRINT #15,"UC";CHR$(18);CHR$(0)

Although, so far I have referred only to the 64 and 1541, the above kernal calls and communcation protocols are analogous to both the Vic-20 with 1540 Disc Drive, and the C128 in 64 mode with 1570/1571 disk drives.

```
LISTEN   = $FFB1  :  TALK    = $FFB4
SECOND   = $FF93  :  TKSA    = $FF96
CIOUT    = $FFA8  :  ACPTR   = $FFA5
UNLSN    = $FFAE  :  UNTLK   = $FFAB
READST   = $FFB7  :  CHROUT  = $FFD2

DEVNUM   = 8
SECADD   = 15
```

```
SNDTBL  .BYTE "M-W",$00,$03,$06,$A9,$31,$8D,$00,
$04,$60
$8D,$00,$04,$60
SNDEND
GETBL   .BYTE "M-E",$00,$03
GETEND
EXTBL   .BYTE "M-R",$00,$04,$01
EXEND

JSR LSTNRT                ; send listen + second addr.
BNE END2                  ; finish if error flag set

LDX #$00
LOOP1 LDA SNDTBL,X        ; get "Memory-Write" string
JSR CIOUT                 ; output byte
INX
CPX #SNDEND-SNDTBL        ; is it done?
BNE LOOP1

JSR UNLSN                 ; send last character, EOI,
and UNLISTEN

JSR LSTNRT

LDX #$00
LOOP2 LDA EXTBL,X
JSR CIOUT                 ; send "Memory-Execute"
                          string
INX
CPX #EXEND-EXTBL
BNE LOOP2
JSR UNLSN

JSR LSTNRT

LDX #$00
LOOP3 LDA GETBL,X
JSR CIOUT                 ; send "Memory-Read"
                          string
INX
CPX #GETEND-GETBL
BNE LOOP3

JSR UNLSN

LDA #DEVNUM
JSR TALK                  ; command drive to talk

LDA SECADD + $60          ; set secondary address
JSR TKSA

JSR ACPTR                 ; input byte
PHA                       ; store it
JSR UNTLK                 ; command disk to stop
                          talking

PLA                       ; retrieve byte
JSR CHROUT                ; print it
END2 RTS

LSTNRT LDA #DEVNUM
JSR LISTEN                ; tell disk to listen

JSR READST                ; get status
AND #%10000000            ; check bit 7 (Device Not
                          Present)
BNE END1                  ; error flag set

LDA #SECADD + $60         ; set secondary address (bit
                          5+6 set)
JSR SECOND

LDA #$00                  ; clear error flag
END1 RTS
```

'CHIP

Eric Doyle takes another long hard look inside your Commodore computer.

THE 6500 SERIES CENTRAL PROCESSOR may be the heart of a computer but it is the slave of the operating system. The OS is either a machine code program or, more usually, the program contained in the Basic ROM. This is the language of convenience devised by Microsoft which the computer uses when it is first powered up.

For Convenience

A language of convenience has nothing to do with the scrawlings on a lavatory wall, it is merely a compromise language used when communications cannot be conducted in the speaker's mother tongue. Basic is such a language, a computer speaks in pure binary but we use English so a stylised form of English has been devised to ease the job of programming. Similarly, a computer does not use variables in its operation, it uses definite memory locations.

The use of variables in Basic is so essential that I used to take it for granted that my old Vic 20 knew what I was talking about when I referred to them. It wasn't long before I started to wonder how it responded to and stored all the variable names I came up with.

One possible answer was that every conceivable variable name had a space allocated by the ROM from the moment I turned the computer on. Of course this is impossible in a mere 3.5K of memory. To allocate sufficient space to 26 strings (A$-X$) would need a reserved space of 6.5K and when arrays are taken into account there wouldn't be enough room in a C128!

The only answer is that the computer adds to the list of variables as each is encountered when a program runs. This still means that space has to be allotted in RAM but it also means that there is a limit to the number of variables which can be defined. It doesn't take much effort to verify this by trying to dimension a large array like DIM A$(256,256).

This list is stored in the same area of memory as the Basic program and effectively reduces the space which is available. The more variables you use, the shorter your program must be.

This means that the computer needs to keep track of where the variable space is to be found. How many types of variable are there? Integer, string and floating point make three but there are also arrays and defined functions.

How do we find where these are stored? Memory locations 45 to 52 hold the secret (C128 owners in C128 mode should add two to all the following values):

Loc1	Loc2	Description
45	46	Start of variables
47	48	Start of arrays
49	50	End of arrays
51	52	String storage
55	56	End of string storage (end of Basic memory)

The locations pointed to by the contents of this tiny block of memory can be found by multiplying the contents of Loc2 and 256 and adding the contents of Loc1.

Down to Work

Time to experiment. Enter the following short program and run it:

```
10 A%=257
```

We must find the start and end of variable storage so type in the following:

```
PRINT 256*PEEK(45)+PEEK(46),256*PEEK
(46)-1)
```

The values obtained will vary depending on the model of your Commodore but you will find the difference between the two values to be seven bytes. This is the amount of space allotted to all integer variables.

Now let's see what's in these seven bytes. Normally we'd use a loop to PEEK each location in turn but this means setting up a new variable and possibly affecting these memory pointers. Luckily, when a variable is set up another pointer pair indicates where the variable is stored. These are locations 71 and 72 (73, 74 in 128 mode). Add the following lines to your program.

```
10 REM BANK1 IN C128 MODE
20 POKE 828,PEEK(71):REM POKE 2816,
PEEK(73) IN C128
30 POKE 829, PEEK(72):REM POKE 2817,
PEEK(74) IN C128
40 LO=PEEK(828)+PEEK(829)*256:REM
2816,2817 IN C128
50 FOR X=0 TO 6:PRINT PEEK (LO+X);:
NEXTX
```

Running the program should give these values: 193,128,1,1,0,0,0.

You're probably wondering what all this means. Let's examine the first two figures. What happens if we subtract 128 from both numbers? We get 65 and 0. Ah! 65 is the ASCII code for the letter A, our variable's name. What if we'd called it AA%? Could it be that we'd have found 193 in the second byte? Try it and see. Change your program line 10 variable to AA%.

Eureka! It works. So we now know that the first two bytes of a stored integer variable is the ASCII code of the first two characters of its name plus 128.

The next two numbers reveal their identity if we treat them in a similar way as we treated the variable pointers earlier. This time we must multiply the first number by 256 and add the second number. Put your computer down, you can do this one in your head. Ok Einstein, the answer is 257. Well, well, this is the value of the variable. Clever isn't it?

What do the zeros mean? Precisely that, nothing whatsoever! They're just padding and you'll see why later.

Complex Integers

Let's try a larger value for AA%, like 32768. See, your computer can't take it. Now try 32767. It likes that, why?

To answer that we've got to go binary. Each byte consists of eight switches which can either be off or on and each indicates a different number. From left to right these are 128, 64, 32, 16, eight, four, two, one. The number four would be represented by the four switch being turned on and all the others off, five would need switch one and four on and 255 would mean all switches were on.

We have found that the stored number is held in two bytes. In this case the 16 switches add eight higher levels to the existing group; 32768, 16384, 8192, 2048, 1024, 512, 256.

Adding all 16 numbers together we get a maximum value of 65535. Subtracting the highest permissable integer, 32767, from this gives 32768, the highest switch value. If this switch is not used for number storage what purpose does it have?

If the switch is on, the integer is a negative value and if it is off, the number is positive.

The proof of the pudding etc. Give AA% a value of -257. The resulting values are 254 and 255. This gives 65279. We know the number is negative so 32768 can be subtracted to give the answer 32511. If 32768 is subtracted again the result is -257!

CHAT' [redacted]

This is a mathematical solution. What actually happens is that a negative number is represented in a form known as two's complement.

To get a two's complement number we need to look at the binary representation of the positive value 257. An on switch is represented by one and off is shown as zero. From our previous example we can see that 257 consists of 256+1, so we turn on those switches only:

0000000100000001

Next we turn all on switches off and all off switches on:

1111111011111110

The final act is to turn the last switch back on:

1111111011111111

Evaluating this gives the value 65279. Now try for yourself to split the 16 switches into two groups of eight and see what values you get using the 128, 64, 32, 16, eight, four, two, one series in both cases. You should get 254 and 255.

What would happen if the last switch was on as in this case?

```
Number     0010011100000100
Complement 1101100011111011
                 +        1
```

We can't turn the last switch on to complete our two's complement so we turn it off and try the next switch. It's also in the on position. Turn it off and move along the line until you find a switch in the off position, remembering to turn off any switches you have to pass.

In the example the third switch is off so we need search no further:

Two's comp. 1101100011111100

To convert the number back to a 'real' value, reverse the process.

```
  0010011100000011
 +               1
  001001110000010
=00100111 00000100
= 39*256   + 4
= 9988
```

Don't forget the minus: -9988

Play with the program giving AA% different identities and then trying work out from the dated PEEKed the number you first thought of.

Alternatively, if all this binary is too much for you, work out the value stored and subtract 65536 from the result to give the negative value.

Highly Strung

Flushed with success, we'll now look at strings. Change line 10 to A$="FIND IT" and run the program again.

This time the first three values are 65, 128, seven. The next two numbers will vary from machine to machine but the last two will both be zeros. Ignore the zeros, they're more padding. This time the first letter of the variable name is directly represented by its ASCII value but the second letter still has 128 added. The next value is interesting count the number of letters in the string. Ah so!

Treat the fourth and fifth figures as a two byte number and use this equation:

```
PRINT CHR$(PEEK(fourth)+PEEK(fifth)
*256)
```

Substitute the values from your program for the words 'fourth' and 'fifth'.

It gives an "F". Enter the line again but put 1+ between the first bracket and the command PEEK. I think we've found it!

Note that the computer reads the value from the program line. Why do locations 51 and 52 point to 40960?

Replace line 10 with this:

```
10 A$="FIND IT":B$= "YOURSELF":C$
A$+B$
```

Run the program and, in the words of the variable, find it yourself.

Functional Storage

Straight in the deep end with you:

```
10 DEF FN AA(A)=PEEK(71)+256*PEEK(72)
20 LO=FN AA(0)-7
```

Erase lines 30 and 40 and run the program.

This time the first character is ASCII+128 and the second is straight ASCII. The third number plus the fourth multiplied by 256 gives the memory location which follows DEF FN AA(A)= in your program. PEEK it to make sure.

In the definition we created a variable A as well as a function variable AA and the location of this variable is given by the next two bytes. As usual ignore the sixth number which is another zero.

Pause for a while to appreciate the cleverness of the program which uses the function definition to find the location of variable A and then backtracks seven places to get to the function entry. PRINT LO+7 should give the same value as the fifth figure plus the sixth figure multiplied by 256.

In my next article I will be raising arrays and immersing you in floating point variables.

Format of Integer Variables

Byte	Contents
1	ASCII value of first character of name + 128
2	ASCII value of second character of name + 128
3	High byte of integer value
4	Low byte of integer value
5	Not used
6	Not used
7	Not used

Integers range from 32767 to -32768

Format of String Variables

Byte	Contents
1	ASCII value of first character of name
2	ASCII value of second character of name + 128
3	Number of characters in string
4	Low byte of string storage address
5	High byte of string storage address
6	Not used
7	Not used

String variables can be up to 256 characters long

Format of defined functions

Byte	Contents
1	ASCII value of first character of name + 128
2	ASCII value of second character of name
3	Low byte of pointer to function's location
4	High byte of pointer to function's location
5	Low byte of pointer to function's internal variable
6	High byte of pointer function's internal variable
7	Note used

Size of function is unlimited

Get on line with David Janda.

COMMUNICATION CORNER

IT'S BEEN PRETTY BUSY IN THE WORLD of comms this past month. There are changes at M'Net and Prestel as well as CNET. So, instead of a particular theme I'll get straight down to the news — read on!

Prestel

It is not often that Prestel introduces major new features on the database, and I don't apologise if that sounds a bit harsh, but I have managed to piece together the following: In a recent Celebrity Chatline interview on M'Net the boss of Prestel Microcomputing indicated that Prestel is looking into the area of keyword searching. Keyword searching is a method of finding a particular bit of info by using — you've guessed it — a keyword. Thus, to find all info on Commodore, I would use 'Commodore' as my keyword.

Just how this is going to be implemented and how much it's going to cost the punter (if anything) is not known at the time of writing.

Another interesting development is File Box. This is a method of sending and retrieving files via the Prestel mailbox. To use this feature, a special bit of software must first be downloaded which is the File Box manager. The scheme is still in its experimental stages, and the File Box software is currently only available for the BBC micro. But, if File Box works well, I should imagine that us Commodore folk will have File Box software for the 64 made available.

The very interesting point about File Box is that 1200/1200 baud transfer is being considered. Now Viewdata systems such as Prestel operate at 1200/75 baud, which means any files that are uploaded are done so at 75 baud (à la Compunet). This is horrendously slow, but being able to upload at 1200 baud, well...!

Micronet

The Net's utterly amazing mega-star PR manager Peter Probert (cue fanfare) gave me a two-page article which outlines the Net according to Simon 'Biggles' D'Arcy, now Publisher of Micronet, and, together with the latest happenings on the Net, it makes interesting reading.

The two important points are: 1) Micronet is going to develop new services this year with the emphasis on more 'serious' applications, and 2) there will be a definite move away from exclusively computer orientated editorial.

The second point is the most important, and already noticeable At the time of writing there is the Sunday supplement, Buttons, and already there is advertising appearing on the Net which has got nothing to do with micros. Now, this begs the question: 'Should MICROnet be doing this?'. Personally, I think it's quite refreshing, but some people have complained, saying that they joined M'Net for micro info, and not for film reviews etc. Fair comment.

It is obvious to see why the Net has adopted this philosophy — money. D'Arcy wants to attract a greater subscriber growth, and including non-micro features would, I think attract a greater audience. Will this formula work? We'll have to see.

Meanwhile, Multi User Dungeon (MUD) should shortly be appearing on Micronet. This is the Net's own version of MUD. It should be quite interesting because it is (to the best of my knowledge) going to be displayed in the Prestel style format.

The Big CNET Move!

By the time you read this, Compunet wil have undergone a major bit of re-organisation. For those of you who are not on Compunet, let me explain. CNET is organised in a directory structure. One directory may be dedicated to music, another graphics and so on. This structure has not changed in a long time, but it should be a different situation by now.

Basically, this re-organisation involves a general tidying up and pulling together of various sections; the objective being to make things easier to find. All software — computing, comms and so on is now in the Software Park. Club 128 incorporates all computing and business stuff, while Live Wire is where you'll find party, chat, news and other features. The move was being made as Your Commodore went to press, and I must say it looks a lot better - nice one CNET!

A few other CNET snippets are; the Radio Amatures CUG is now open to all, and more new features are on their way. CNET's Editor Jane Fairbank wouldn't give me any more info, but I understand a new adventure/role playing game is being considered.

Snippets

Wow, I'm running out of room! OK, here we go: Two new Multi User Games (MUGs) are under beta-testing. The first is called 'Gods' which represents a world based on a North African Sea Port (!). Registration is 23 quid a year. More info from Lap of The Gods Ltd, 166 Portobello Road, London W11 2EB.

Another new MUG is from Dreygun ltd. The game is styled on the classic MUD, but has different treasures and locations with eight distinct 'them' areas, each with about 2000 locations! £20 will get you a starter pack which includes map, hints, ID and password and 30 hours of playing time. Write to Dreygun Ltd, 30 Warren Drive North, Tolworth, Surrey KT5 9L6 for more info.

Logoff

That's it for now folks. As you can see from my contact list I am now on One-To-One and Telecom Gold/Microlink. Details of these and other info next month. Till next time!

David Janda can be contacted on the following services:
Prestel/M'Net — **919992677**
Compunet — **D. JANDA**
Gold/Microlink — **72:MAG95307**
One-To-One — **13419001**

ASSEMBLER 128

Steve Carrie brings you an Editor Assembler for your C128.

THIS UTILITY IS INTENDED for use on a Commodore 128 system operating in 128 mode. The program is fairly simple as assemblers go, but it could be useful to someone who perhaps cannot afford a more comprehensive package. Both tape and disk are supported as well as a printer.

Before going on to describe the program in more detail, I will give a brief overview.

In 128 mode, the computer maintains two 64K banks of RAM (RAM 0 and RAM 1). There are a couple of common areas in the memory map to allow programs to operate correctly between banks. There are 16 predefined memory configurations. Of these, BANK 12 is used for this program. This is a combination of RAM 0 from 0400-7FFF hex and the kernal ROM from B000 hex upwards. This allows the program to make direct calls to the kernal routines without having to go through a complicated bank switching routine. This area from 0000 to 03FF hex is a common area in all banks. The source code is edited in RAM 1 from 04000-FEFF hex (approx. 62.75K) although the actual amount allocated may be altered.

The assembler will allow code to be assembled in any of the predefined banks (actually only RAM 0 and RAM 1 are valid unless you have a 256K machine). and a limited relocation facility is provided. The Commodore machine language monitor (MLM) can be accessed at any time from the editor command mode. The MLM X command will re-enter the editor.

The program will be given in two sections. In this article I will deal with the editor. When you have typed in the editor you may use any of the commands except C which calls the assembler. Without the assembler code the system will probably crash.

The Editor

First type in and save the small program RELOCATE. This program moves Basic to a higher address in RAM 0 so as to allow the assembler to be constructed in the correct place.

Note

You must always run this program before using the LOADER.

Now type in and save the program LOADER. Use this to enter the hex data from the main listing. It will be a long job so my advice is that you do it in stages. That way you won't get frustrated if you keep making mistakes.

If you look at the main hex dump, you will see that each line has an address, a hex data string and a checksum value. When you run the LOADER, the menu will appear. Options two and three save and load the complete program area i.e. the memory occupied by both the editor and the assembler (which will be published in the near future).

When asked for the start address, enter 1C01. If you save an incomplete version of the program and intend to resume at a later time, make a note of the next address you have to enter and use this when asked for the start address. When you have entered the whole program, type END,0 to return to the menu.

During entry the program uses the checksum to validate the input. It will request re-input if an error is found otherwise it will prompt you with the address of the next data string.

Keep these two programs for use with the assembler.

Running the Editor

The program can be loaded and saved like a Basic program. Type RUN to enter the editor. A message is displayed and the computer locked into lower case mode. All commands must be entered in lowercase although uppercase may be used in source text. The assembler translates everything into lowercase anyway. The exception to this is the text directive which will be explained when you get the assembler.

The Editor in Operation

The operation of the editor is similar to that of the normal Commodore line editor. Lines are entered with line numbers and the cursor keys may be used to move around the screen.

When first starting out on the program, I suggest you think carefully about how big the source code is likely to be and allocate as necessary.

Unlike the Commodore editor, this program does not remove spaces (max line length is 255 characters). This means that you can make your text more readable by indenting sections of code.

A list of error messages follows. Most are self explanatory.

Error Messages

Invalid or badly formatted command.
Invalid sub-command.
Line does not exist.
Invalid or missing parameter.
I/O error.
Invalid/out or range line number.
No source program.
String too long.
Search fails (not really an error).
Out of memory (program too big!).
Search string too long.
Replace string too long.
Illegal device specification.
Cannot access device while printer is engaged.
Not valid command for tape.

The Editor Commands

- **?** Display help message
- **A** Auto line numbering on/off
- **B** Set bottomof text memory in RAM 1
- **C** Compile
- **D** Delete block
- **E** End Edit and exit to Basic
- **F** Display memory allocation
- **I** I/O
- **K** Kill program
- **L** List lines
- **M** Enter monitor
- **O** Recover (old) program
- **P** Printer prefix
- **R** Renumber lines
- **S** Search (and replace)
- **T** Set top of text memory in RAM 1

Editor Commands in Detail

? — Display help page. This displays a summary of the editor commands.

A — Auto line numbering. This enables and disables the automatic line numbering during program entry. The format of the command is A <increment>as in A 10 which sets an increment of 10. Auto numbering will commence from the last line number entered plus the increment. The operation of this command is similar in most respects to the Basic 7.0 auto command.

B — Set bottom of text memory in Ram 1. Initially the allocation to the editor in RAM 1 is about 63KB. This command along with T alters that allocation. Addressing is done in blocks of 256 bytes, numbered 0 to 250. Giving the command B20 will set the bottom of text to block 20 (actual address is 256*(block+4) therefore this address would be 6144 decimel or 1800 hex). The message ARE YOU SURE is printed and the user must give the Y response before the relocation is carried out. This is done since this command destroys any program in memory.

C — Compile. Details will be given with the assembler listing.

D — Block delete. Format is D<start>—<end>as in D 20-230. Deletes a block of lines.

E — End edit and exit. The message ARE YOU SURE is printed and the user must give a Y response. This is donw since exiting the editor may destroy the program in RAM 1.

F — Displays current text memory allocation and number of bytes free.

I Input output. There are several forms of this command.

I — Display I/O information
IC — Display current device directory
ID — set current device
IN — Set current filename
IS — Save file
IL — Load file

An important concept is that of the current device and filename. For example:
Enter ID
Editor responds
CURRENT DEVICE =8 [DISK]
ENTER NEW DEVICE>
Enter 1
Editor responds
NEW DEVICE =1 [TAPE] OK.
Now enter IN
Editor responds
CURRENT FILENAME =""
ENTER NEW FILENAME>
Enter "test" including the quotes
Editor responds
NEW FILENAME ="test" OK.
Now enter 1
The editor prints the following:
CURRENT DEVICE =1 [TAPE]
CURRENT FILENAME ="TEST"
PRINTER IS OFF
When IS or IL is used the current device and name are used. Note that ID and IN may be used as follows:
ID1
IN"test"
The IC command will display the directory of the current device if it is a disk drive.

IS and IL always load and save from/to the current base block as determined by the B command. These two commands will fail if no filename has been set or the printer is on.

K — Kill program. Simply deletes the current file, Confirmation is required. The program may, under certain circumstances, be recovered with the O command.

L — List lines. Format is<start>—<end>. Run/stop may be used to halt the listing.

M — Enter MLM. Monitor's X command will re-enter editor command mode.

O — Recover deleted program.

P — Printer prefix. Prefixing most commands with this will cause output to be deflected from the screen to the printer. Will not work with IL,IS and IC.

R — Renumber lines. Format is R<start>,<increment> as in R 10,20.

S — Search. There are two different format to this command. 1. S "<string1>",R"<string2>":<start>, <end>
2. S "<stringl>":<start>,<end>.
Form 1 finds every occurence of <stringl>and replaces it with string 2 . If * is used for<start>or <end> , scanning will be from/to the beginning/end of the file.
Form 2 finds every occurence of<stringl>.

T — Set top of RAM 1. Same type of parameters as B. Note that bottom cannot be greater than top.

PROGRAM: RELOCATE

```
10 BANK0:POKE16384,0
20 POKE 46,64
30 PRINT "(DOWN)NOW RUN "CHR$(34)"
LOADER"CHR$(34)
40 NEW
```

PROGRAM: LOADER

```
10 DO
20 SCNCLR
30 PRINT "ASEM 128 HEX LOADER"
40 PRINT
50 PRINT "1. ENTER HEX DATA"
60 PRINT "2. SAVE CURRENT WORK FIL
E"
70 PRINT "3. LOAD CURRENT WORK FIL
E"
80 PRINT "4. END"
90 PRINT "(DOWN)PLEASE CHOOSE OPTI
ON"
100 DO:GETKEYA$:A=VAL(A$):LOOP UNT
IL A>0 AND A<5
110 :
120 :
130 ON A GOSUB 1000,2000,3000
140 LOOP UNTIL A=4
150 END
160 :
170 :
1000 REM **************************
****
1010 REM HEX ENTRY ROUTINE
1020 REM **************************
****
1030 :
1040 SCNCLR
1050 :
1060 DO
1070 :    INPUT "ENTER ADDRESS (IN
 HEX)";S$
1080 LOOP UNTIL S$<>""
1090 S=DEC(S$)
1100 :
1110 PRINT "NOW ENTER DATA AS IT I
S PRINTED IN THE  LISTINGS. THE CO
MPUTER WILL PROMPT YOU  WITH THE A
DDRESS."
1120 PRINT "YOU SHOULD ENTER THE D
ATA STRING         FOLLOWED BY A CO
MMA THEN THE CHECKSUM    VALUE AT T
HE END OF EACH LINE."
1130 PRINT "ENTER 'END,0' TO EXIT
TO MENU.
1140 :
1150 PRINT
1160 :
1170 DO
1180 :    DO
1190 :          TT=0:H$="":C$=""
1200 :          PRINT HEX$(S)
1210 :          INPUT H$,C$
1220 :          IF H$="END" THEN EX
IT
1230 :
1240 :          FOR X=1 TO 64 STEP
2
1250 :          AD=(X-1)/2+S
1260 :          BY=DEC(MID$(H$,X,2)
)
1270 :          BANK0:POKEAD,BY
1280 :          TT=TT+BY
1290 :          NEXT
1300 :
1310 :          IF TT<>DEC(C$) THEN
 PRINT "DATA ERROR. RE-ENTER THIS
LINE"
1320 :
1330 :    LOOP UNTIL TT=DEC(C$)
1340 :
1350 :    S=S+32
1360 LOOP UNTIL H$="END"
1370 RETURN
1380 :
1390 :
2000 REM **************************
****
2010 REM SAVE CURRENT WORK FILE
2020 REM **************************
****
2030 :
2040 GOSUB 5000 SETUP
2050 PRINT "SAVING WORK AREA"
2060 POKE 253,1:POKE 254,28
2070 SYS DEC("FFD8"),253,192,62
2080 :
2090 PRINT DS$
2100 SLEEP 2
2110 RETURN
2120 :
2130 :
3000 REM **************************
****
3010 REM LOAD CURRENT WORK AREA
3020 REM **************************
****
3030 :
3040 GOSUB 5000 SETUP
3050 PRINT "LOADING WORK AREA"
3060 SYS DEC("FFD5"),0,1,28
3070 :
3080 PRINT DS$
3090 SLEEP 2
3100 RETURN
3110 :
3120 :
4000 REM **************************
```

```
*****
4010 REM COLLECT DEVICE
4020 REM *************************
*****
4030 :
4040 INPUT "DEVICE NUMBER ";D
4050 INPUT "FILENAME ";A$
4060 RETURN
4070 :
5000 REM *************************
*****
5010 REM SETUP FOR LOAD/SAVE
5020 REM *************************
*****
5030 :
5040 GOSUB 4000
5050 BANK 12
5060 SYS DEC("FFBA"),1,D,0
5070 :
5080 SYS DEC("FF68"),0,1
5090 :
5100 BANK 1
5110 PT=POINTER(A$)
5120 L=PEEK(PT):LO=PEEK(PT+1):HI=P
EEK(PT+2)
5130 BANK 12
5140 SYS DEC("FFBD"),L,LO,HI
5150 RETURN
```

PROGRAM: MAIN LISTING

```
1C01 101C0A00FE0231323A9E37313837000000004CB3234C1324080053303A000000 05B2
1C21 0000000000000000000000000000000000000000000000000000000000000000 0000
1C41 0000930E0B0D2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A 04FD
1C61 2A2A2A2A2A2A2A2A2A2A2A2A2A0D6153454D203132382E0D615353454D424C59 06EB
1C81 204C414E4755414745204544449544F522F434F4D50494C45522E0D634F4D4D4F 08AB
1CA1 444F5245203132382056455253494F4E2E0D615052494C20313938362073E64 081B
1CC1 2E632E0D615353454D424C4552202F20454449544F522056455253494F4E2031 0857
1CE1 2E300D2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A 052D
1D01 2A2A2A2A2A2A2A2A2A2A0D0D00930D0C8E454E44204F4620454449544F522045 06D0
1D21 5845435554494F4E2E0D003F4144454B4C5253494F50434D425446008827D927 08F2
1D41 09279227FB26A92561261C29DD2CFA2D632E3E302F2F7A2FA92F04300D415245 09CB
1D61 20594F552053555245202859 2F4E293F000D454449544F5220434F4D4D414E44 07F5
1D81 533B0D4C205B4E5D5B2D4E5D20204C495354204C494E45530D52205B4E5D5B2C 0863
1DA1 495D20205245 4E554D424552204C494E45530D41205B495D2020202020204155 07A6
1DC1 544F204E554D424552494E47204F46462F4F4E0D44205B4E2D4E5D2020202044 07E7
1DE1 454C455445204C494E45530D3F202020202020202020202048454C502028544849 06EC
1E01 53204D455353414745290D4520202020202020202020202045584954 0D4920202020 0643
1E21 202020202020492F4F20494E464F524D4154494F4E0D4944205B4E5D20202020 072D
1E41 205345542043555252454E5420492F4F20444556494345 0D494E20223C535452 0821
1E61 3E222053455420435552 52454E5420492F4F2046494C454E414D450D49532020 07E0
1E81 2020202020202053415645204355 5252454E542046494C450D494C2020202020 06D4
1EA1 2020204C4F4144204E45572046494C450D4943202020202020202020202044 4953 0683
1EC1 504C4159204449534B2044495245435 44F52590D4949202020202020202020 2049 075E
1EE1 4E495449414C495345204355 52 52454E5420492F4F20444556494345 0D4B2020 0834
1F01 20202020202020204B494C4C202844454C4554452920435552 52454E54204649 0742
1F21 4C450D4F2020202020202020202020205245434F564552204120 4B494C4C45442050 06E9
1F41 524F4752414D0D53223C535452494E473E225B2C52223C535452494E473E225D 0888
1F61 3A4E5B2C4E5D0D2E2020202020202020202020205345415243482028262052 45504C 06AC
1F81 414345290D4620202020202020202020202044495350 4C4159204D454D4F52592053 0707
1FA1 54415455530D502020202020202020202020454E41424C45205052494E5445520D 0726
1FC1 432020202020202020202020434F4D50494C45202841535353454D424C45290D4D20 06C3
1FE1 20202020202020202020454E544552204D4F4E49544F520D422020202020202020 0635
2001 2020534554205441585420 4D454D4F5259204241534 50D542020202020202020 0726
2021 2020534554205445585420 4D454D4F525920454E440D00494E56414C4944204F 0804
2041 52204241444C5920464F524D415454454420434F4D4D414E4400494E56414C49 0886
2061 44205355422D434F4D4D414E44004C494E4520444F4553204E4F542045584953 0857
2081 5400494E56414C4944204F52204D495353494E4720504152414D455445520049 0860
20A1 2F4F204552524F5200494E56414C49442F4F5554204F462052414E4745204C49 084D
```

```
20C1 4E45204E554D424552004E4F20534F555243452050524F4752414D0053545249 0884
20E1 4E4720544F4F204C4F4E47005345415243482046414C53004F5554204F4620 0809
2101 4D454D4F5259005345415243482053545249 4E4720544F4F204C4F4E47005245 087F
2121 504C414345205354524 94E4720544F4F204C4F4E4700494C4C4547414C204445 0861
2141 56494345205350454349464943415449 4F4E0043414E4E4F5420414343455353 089D
2161 20444556494345205748494C4520505249 4E544552204953202020454E474147 082B
2181 4544004E4F542056414C494420434F4D4D414E4420464F52205441504500 3820 07D2
21A1 5B206F208320A020AA20CB20DD20ED20FA2008211F21372154218421F600D002 0AE9
21C1 F60160A9FFD600D500D002D6016078A000A986A2888DAA028EB902A20120D3F7 0F33
21E1 A20120DAF75860A58A38E58485FDA58BE58585FE6085FB86FC20732220E821A5 11A0
2201 8A85861865FB858A8588A58B858765FC858B858920CF21A28620C421A28820C4 102F
2221 2120CF21A2FD20C421A5FED0EAA5FDD0E66085FB86FC20E821A584858638E5FB 1361
2241 8588A5858587E5FC858920CF21A28620BD21A28820BD21A2FD20C421A5FED0EA 1191
2261 A5FDD0E6A58A38E5FB858AA58BE5FC858B60A58A1865FBAAA58B65FCCD431C90 13CD
2281 0FF0034C8C22EC421CB0F8A2094CEE2460A2008616861720F122B039290F48A5 0C73
22A1 1748A5164806162617B02B06162617B0256818651685166865178517B0180616 07BE
22C1 2617B012681865168516A5176900851720EB2290C560A2054CEE2478A20148A9 0B49
22E1 828DB9026820DAF75860E63DD002E63EA000B13DC93AB00AC920F0EE38E93038 0F8F
2301 E9D06078A201A98220D0F7586078A20148A9828DC8026820E3F75860A547A448 0FD5
2321 8582848318A001200423F027A004C8200423D0FAC89865828584A0000820DC22 0CB8
2341 28A5836900C80820DC2228A6848682858390D260A547A648A00185828683200 4 0D7A
2361 23F027C8C8A517200E23901FF00388D00AA51688200E239012F0108820042385 0B65
2381 8488200423A68438B0CE1860A20020CFFFC90DF00B9D000EE8D0F3A2074CEE24 0E69
23A1 A9009D000EA900A20E853D863EA90D4CD2FFA53D48A53E48BA8E341CA906A20F 0C53
23C1 8D20D08E21D085F1A943A01C201125A9008D401CA9FA8D411CA900A204854786 0D30
23E1 48A9FFA2FE8D421C8E431CA547A64885828683186902858A8A6900858BA00098 0E1F
2401 20DC22C820DC22A9002090FF8D361C8D351CAE341C9AAD301CF003208D2EA920 0C41
2421 8DFA22AD361CF008AD351CF00320AB27208D2320F122F0DA9021A200DD2C1DF0 0DB9
2441 0AE8E01090F6A2004CEE248A0AAABD3D1D8516BD3E1D85176C1600A9FF8DFA22 0DDF
2461 20922205523 0820F12228901E48A004200423C8C900D0F8981865828584A583 0BD1
2481 6900858598A200203322688D351CC900F050A204A000B13DF004E8C8D0F8E88A 0E83
24A1 8D331CA200A4828484A483848520F621A000A9FF20DC22C820DC22C8A51620DC 0EDE
24C1 22C8A51720DC22C88484A0008485A485B13DC88485A48420DC22C88484CC331C 0F56
24E1 D0EC201D23A9008D2F1C4C13248A0A48A90D20D2FFA90720D2FF68AABD9F2148 0D1B
2501 BDA021A868201125A90D20D2FF4C132484178516A000B116F00A20D2FFC8D0F6 0E24
2521 E617D0F260A547A64885828683A001200423F00160A2064CEE2485168417204 9 0CE7
2541 25A900A00E4C1125A00084888489A200A51638F9A1258516A517C8F9A1259007 0C80
2561 8517E8884C512588A51679A12585168AD0062488300610 12A2808688093086FD 0C3B
2581 A6899D000EE88689A6FDC8C8C00890BEA5160930A6899D000EE8A9009D000E60 0E84
25A1 1027E80364000A0020262520EB22F04820922220552320F122F032C92DF0034C 0A56
25C1 5C2620EB22F01D209222A58248A58348205523A5828586A583858768858368 85 0DCA
25E1 824C0426A9FF858785864C0426A5828586A58385874C0426A547A64885828683 0DF3
2601 4CE525A001200423F02D203B2620E1FFF025A583C587900DF002B01BA582C586 0E31
2621 90034C3826A00020042348C8200423858368858 24C04264C1324A00220042348 081C
2641 C8200423A868203B25A004200423F00620D2FFC8D0F5A90D4CD2FFA2034CEE24 0DD4
2661 20262520EB22F02B20922220F122F02BC92CD06E20EB22F069A51648A5174820 0CB5
2681 9222A5168586A517858768851768851 64CA426A90A8516A9008517A90A8586A9 0C59
26A1 008587A547A64885828683A001200423F02DC8A51620DC22C8A51720DC22A000 0CDE
26C1 A5861865168516A58765178517B01620042348C82004238583688582 4CAC264C 0AE2
26E1 1324A2032CA2054CEE24A95DA01D201125208D2320F122C959602026 2520EB26 0A47
2701 D0034CEC234C13242026 2520EB22F06C20922220552390 6720F122C92DD05D20 0B7E
2721 EB22F058A5828586A583858720922220552390 4BA00020042348C82004238583 0C48
2741 688582A583C5879039F002B008A582C586902FF02DA58238E58648A583E58748 1092
2761 A682A58686848582A683A58786858368AA68203322201D234C1324A2032CA2 0D11
2781 022CA2054CEE24A972A01D2011254C132420EB26D011AE341C9A68853E68853D 0AE3
27A1 A90EA01D4C11254C1324A516186D371C8516A5176D381C8517A51648A5174820 08E7
27C1 4925688517688516A200BD000EF0069D4A03E8D0F586D06020EB22D0098D361C 0D0A
27E1 8D351C4C1324209222A516A6178D371C8E381CA9018D361CA9008D351C4C1324 0902
2801 A00085828683B13DC922F00A9182C8D0F5A20A4CEE24C000F00160A2004CEE24 0F3E
2821 18653D853DA53E6900853E6020EB22C922F0034C142920EB22A900A20C200128 0A4C
2841 C0A09005A20A4CEE248C3A1CC89820212820F122C93AD008A9008D391C4CA628 0C5D
2861 A2018E391CC92CF0034C142920EB22C952F0034C142920EB22C922F0034C1429 0B4F
```

```
2881 20EB22A900A20D200128C0A09005A20B4CEE248C3B1CC89820212820F122C93A 0BB0
28A1 F0034C142920EB22D0034C1429C92AF00A20922205523905DB00BA547A44885 0B63
28C1 82848320EB22A5168588A517858920F122F035C92CD03C20EB22F037AAA58248 0F0E
28E1 A58348E02AF00A209222055239027B006A9FF85168517A516A6178D371C8E38 0C4A
2901 1C68858368858260A5888D371CA5898D381C60A2002CA2024CEE24A9008D361C 0C2F
2921 8D351C202D28A0002004239900 0BC8C00490F5A2008E3F1C8E3D1C200423990 0941
2941 0BF009EE3D1CEE3F1CC8D0EFEE3F1CA904A20B853D863EA2008E3E1CAE3D1CEC 0DC1
2961 3A1CB0034C752AA000B13DD9000CD008C8CC3A1C90F3B00BCE3D1CEE3E1CE63D 0CF9
2981 4C5D298C3C1CAE391CD0034C4F2AA200BD000B9D000EE8E00490F5A200EC3E1C 0B9F
29A1 F009BD040B9D040EE8D0F2A000A90E853EB9000D913DC8CC3B1C90F59818653D 0D8E
29C1 8586A53E8587A53D186D3A1C853DA90B853EA000B13D9186F003C8D0F7AD3F1C 0E1F
29E1 186904481865828584A583690085856 8A200203322A204BD000EF003E8D0F8E8 0CEB
2A01 8AA200A4828484A483848520F621A000B9000E20DC22C8C00490F5B9000E0820 0D46
2A21 DC2228F003C8D0F3A58248A58348201D23AD020EAC030E203B25A92020D2FFA9 0D40
2A41 04A00E201125A90D20D2FF189020A58248A58348AD020BAC030B203B25A92020 0A33
2A61 D2FFA904A00B201125A90D20D2FF688583688582AD020BCD371CD008AD030BCD 0D3F
2A81 381CF019A00020042348C8200423858368858 2A001200423F0034C2729AD3C1C 098E
2AA1 F0034C1324A2084CEE24C901F00DC9089004C90C9005A20C4CEE246020F122C9 0C7C
2AC1 22D01620EB22A91EA21C20DF2AC010B0088C1A1CA900918260A2034CEE24A000 0BEC
2AE1 85828683B13DC922F018C900F0149182C8C01090EFA2072CA203A9008D1E1C4C 0E1E
2B01 EE24C000F0F260AD191CC901D00160AAA90FA820BAFFA20CA9012068FFAE1A1C 0E97
2B21 E8E8E88AA21BA01C20BDFF20C0FFB00520B7FFF01220E7FFAD191CC901F00320 10C8
2B41 412EA2044CEE24600D43555252454E5420444556494345 3D000D454E54455220 08BB
2B61 4E4557204445564943453E000D43555252454E542046494C454E414D453D2200 07E8
2B81 454E544552204E45572046494C454E414D453E00204F2E4B2E0D004E45572044 0798
2BA1 4556494345 3D004E45572046494C454E414D453D22000D4C4F4144494E472046 07C4
2BC1 494C452022000D46524F4D20444556494345 20000D534156494E472046494C45 0762
2BE1 2022000D544F20444556494345 2000205B544150455D00205B4449534B5D0050 0737
2C01 52494E544552204953200 04F46460D004F4E0D00AD191C48A000203B2568C901 07BE
2C21 F007A9F8A02B189004A9F0A02B4C1125A91EA01C201125A9224CD2FFA949A02B 0D73
2C41 2011254C152CA96DA02B20112520312CA90D4CD2FFA900A02C201125AD301CD0 09FE
2C61 07A90CA02C4C1125A911A02C4C112520EB22D012203D2CA95AA02B201125208D 097B
2C81 2320F122F020209222A517D01CA51620AB2A8D191CA99CA02B20112520152CA9 0AC4
2CA1 95A02B2011254C1324A20C4CEE2420EB22D00D20472CA981A02B201125208D23 09FD
2CC1 20BD2AA9A8A02B20112520312C4CA02C203D2C20472C20562C4C132420EB22D0 094C
2CE1 034CD12CC944D0034C702CC94ED0034CAF2CAE301CF005A20D4CEE24C953D003 0D10
2D01 4C1C2DC94CD0034C8D2DC943D0034C9B2EC949D0034C232F4C1324202625AD1E 0AB3
2D21 1CD005A2034CEE2420082BAD191CC901D00320E9E9A9D5A02B20112520312CA9 0B7D
2D41 E4A02B20112520152CA90D20D2FFA00120702DA547A44885168417A916A68AA4 0C0C
2D61 8B20D8FF20872DA90D20D2FF4C1324A901A20C2068FFAE191C20BAFFAD1A1CA2 0DA0
2D81 1EA01C4CBDFFB001604C362BAD1E1CD005A2034CEE24A547A44885828483A001 0CE6
2DA1 200423F00520EB26D03AAD191CC901D00320C8E9A9B7A02B20112520312CA9C7 0C35
2DC1 A02B20112520152CA00020702DA900A647A44820D5FFB00F868A848BA90D20D2 0BDB
2DE1 FF201D234C1324A547A44885828483A900A00120DC224C362BA547A448858284 0C40
2E01 83A001200423F00AA92DA02E2011254C1324A001A9FF20DC22201D23A5821869 0A51
2E21 02858AA5836900858B4C13240D50524F4752414D2052455349444 54E540D0049 092E
2E41 A90FAE191CA820BAFFA901A20C2068FFA901A240A02E20BDFF20C0FFA90F20C3 0EAB
2E61 FF60AD301CD025A904AAA00720BAFFA90020BDFF20C0FFB010A9018D301CA204 0E71
2E81 20C9FF20EB224C34244C132420CCFFA90420C3FFA9008D301C60A901AE191C20 0C45
2EA1 192FA00020BAFFA901A20C2068FFA901A218A02F20BDFF20C0FFB04920B7FFD0 0F2D
2EC1 44A20120C6FF20092F20092F20092F20092FF02620092F20092F20092FF0EDC9 08F0
2EE1 22D0F720092FF00AC922F0F720D2FF4CE42EA90D20D2FF4CCD2E20CCFFA90120 0FFF
2F01 C3FF4C13244C362B20CFFFB0F84820B7FF29BFD0F0686024E001F00160A20E4C 0F68
2F21 EE24AE191C20192F20412E4C1324A948A22FAC000A8C311CAC010A8C321C8D00 08DE
2F41 0A8E010A4C00B0A9068D00FFAD311CAE321C8D000A8E010AA963A02F2011254C 097D
2F61 13240D654449544F522F415353454D424C455220312E300D0020EB22C900F023 08BD
2F81 209222A517D01C20D42FC9FBB015CD411CF002B00E8D401C6904A20085488647 0C94
2FA1 4CEC23A2034CEE2420EB22C900F0F4209222A517D0ED20D42FC9FBB0E6CD401C 102B
2FC1 90E18D411C186904A2FF8D431C8E421C4CEC23A5164820EB26F0034C13246860 0C26
2FE1 544558542053504143452 03D0020424C4F434B532C0020425954455320465245 07DC
3001 450D00AD411C38ED401C18690148A9E1A02F20112568A000203B25A9EEA02F20 0A64
3021 1125AD421C38E58A48AD431CE58BA868203B25A9F7A02F2011254C1324202625 0AEF
```

GO FASTER 64

Tony Crowther shows how to speed up the C128 in C64 mode.

WHEN COMMODORE launched the C128 computer they said that it had a C64 computer inside it. They claimed at the time that this was completely compatible with the normal C64 computer. Well, time has demonstrated that it isn't, as many programs will not work on the C128. Obviously there must be some differences between the normal C64 and the one in the C128.

The differences don't only have to be a bad point. It is possible to use some of them to your advantage if you know what you are doing.

As you probably know the C128 has a FAST instruction that blanks the screen and causes it to run in 2MHz mode. You probably didn't know that this fast mode is also available from the C64. Below are two example programs that will allow you to use this 'bug' to your benefit. For each program I have supplied an assembly listing for those of you who wish to know how they work. I use the Machine Lightning assembler, but for those who simply wish to use them I have included a simple Basic loader that will SAVE the machine code on to your disk or tape. If you are using tape then change the ,8,1 after the SAVE"name" in each loader to ,1,1.

Program 1

This machine code program when activated by SYS 49152 will access the C128's 2MHZ processor so that the execution speed of the C64 is changed from .9Meg to 1.2Meg with full screen display. The program doesn't finish there however. By pressing the 'F7' key the screen is blanked out from the bottom up. By blanking more of the screen it is possible to alter the speed of the C64 from 1.2Meg up to 2Meg. The greater the amount of screen blanked the faster the program. The screen can be unblanked by pressing the 'F8' key slowing the C64 down.

Note when this program is running you will lose the ability to talk to any peripherals. Therefore hit RUN STOP/RESTORE to quit. If at this point the screen goes funny then either hit the RESET button or type the following line in blind:

POKE 53296,252.

RUN STOP/RESTORE does not always reset the 2Meg latch as this is inside the C128 and not the C64, hence the need for this POKE to reset the latch to normal.

Both of the programs in this article can be used on the 128 if you alter the interrupt vectors at $314 and $315. Don't attempt this unless you know what you are doing.

Program 2

This program works in a similar fashion to the above program. However, this time the screen is blanked from the top down, not from the bottom. Blanking in this program is also much quicker than the other version of the program.

So what use are these programs? You could use the routines to improve the speed of calculation programs. Who knows we may even start to get programs that have a faster mode for C128 computers.

PROGRAM: 1 CODE

```
 10                      *=$C000
 20 SHIFT               =1
 30 SHIFTKEY            =653
 40 CHECKKEY            =197
 50 FUNCTION7           =3
 60 RASTER              =53266
 70 SCREENBLANK         =53265
 80 BLANK               =123
 90 NORMAL              =27
100 SCREENTOP           =50
110 MEGABIT             =$D030
120 ON                  =$FD
130 OFF                 =$FC
140 NORMIRQ             =59953
150 LEAVEIRQ            =60033
160 JMPVECIRQ           =788
170 TIMERA              =56334
180                      ;
190                      ;
200                      .ORG     $C000
210                      ;
220 START                SEI
230                      ;
240                      LDA #IRQ&255
250                      STA JMPVECIRQ
260                      LDA #IRQ/256
270                      STA JMPVECIRQ+1
280                      ;
290                      LDA #0
300                      STA TIMERA
310                      ;
320                      LDA #NORMAL
330                      STA SCREENBLANK
340                      ;
350                      LDA #1
360                      STA 53273
370                      STA 53274
380                      ;
390                      LDA #250
400                      STA MOVINGRAST
410                      ;
420                      CLI
430                      ;
440                      RTS
450                      ;
460                      ;
470                      ;
480 IRQ                  LDA #1
490                      STA 53273
500                      ;
510                      INC COUNT
520                      LDA COUNT
530                      BNE IRQPART2
540                      ;
550                      LDA #NORMAL
560                      STA SCREENBLANK
570                      ;
580                      LDA #OFF
590                      STA MEGABIT
600                      ;
610                      LDA MOVINGRAST
620                      STA RASTER
630                      ;
640                      PLA
650                      TAY
660                      PLA
670                      TAX
```

```
680                 PLA
690                 RTI
720                 ;
730 IRQPART2        LDA #BLANK
740                 STA SCREENBLANK
750                 ;
760                 LDA #ON
770                 STA MEGABIT
780                 ;
790                 LDA #SCREENTOP
800                 STA RASTER
810                 ;
820                 LDA #255
830                 STA COUNT
840                 ;
850                 JSR PRESSF7
860                 ;
870                 JMP NORMIRQ
900                 ;
910 PRESSF7         LDA CHECKKEY
920                 CMP #FUNCTION7
930                 BNE NOKEYPRESS
940                 ;
950                 LDA SHIFTKEY
960                 AND #SHIFT
970                 BEQ NOSHIFTKEY
980                 ;
990                 LDA MOVINGRAST
1000                CMP #58
1010                BEQ ERROR
1020                ;
1030 LOOP1          DEC MOVINGRAST
1040                JSR CHECKFORDMA
1050                BCS LOOP1
1060                ;
1070 ERROR          RTS
1080                ;
1090                ;
1100 NOSHIFTKEY     LDA MOVINGRAST
1110                CMP #250
1120                BEQ ERROR
1130                ;
1140 LOOP2          INC MOVINGRAST
1150                JSR CHECKFORDMA
1160                BCS LOOP2
1170                ;
1180 NOKEYPRESS     RTS
1190                ;
1200                ;
1210                ;
1220 CHECKFORDMA    LDA MOVINGRAST
1230                AND #7
1240                CMP #3
1250                BNE NODMA
1260                ;
1270                SEC
1280                RTS
1290                ;
1300 NODMA          CLC
1310                RTS
1320                ;
1330                ;
1340 MOVINGRAST     .BYTE   0
1350 COUNT          .BYTE   0
1360                ;
1370                ;
1380 END            .END
```

PROGRAM: 2 CODE

```
10                  *=$C000
20 SHIFT            =1
30 SHIFTKEY         =653
40 CHECKKEY         =197
50 FUNCTION7        =3
60 RASTER           =53266
70 SCREENBLANK      =53265
80 BLANK            =123
90 NORMAL           =27
100 SCREENBOTTOM    =250
110 MEGABIT         =$D030
120 ON              =$FD
130 OFF             =$FC
140 NORMIRQ         =59953
150 LEAVEIRQ        =60033
160 JMPVECIRQ       =788
170 TIMERA          =56334
180                 ;
190                 ;
200                 .ORG    $C000
210                 ;
220 START           SEI
230                 ;
240                 LDA #IRQ&255
250                 STA JMPVECIRQ
260                 LDA #IRQ/256
270                 STA JMPVECIRQ+1
280                 ;
290                 LDA #0
300                 STA TIMERA
310                 ;
320                 LDA #NORMAL
330                 STA SCREENBLANK
340                 ;
350                 LDA #1
360                 STA 53273
370                 STA 53274
380                 ;
390                 LDA #1
400                 STA MOVINGRAST
410                 ;
420                 CLI
430                 ;
440                 RTS
450                 ;
460                 ;
470                 ;
480 IRQ             LDA #1
490                 STA 53273
500                 ;
510                 INC COUNT
520                 LDA COUNT
530                 BNE IRQPART2
540                 ;
550                 PHA
560                 PHA
570                 PHA
580                 PLA
590                 PLA
600                 PLA
610                 ;
620                 LDA #OFF
630                 STA MEGABIT
640                 ;
650                 LDA #SCREENBOTT
660                 STA RASTER
670                 ;
680                 LDA #NORMAL
690                 STA SCREENBLANK
700                 ;
710                 PLA
720                 TAY
730                 PLA
740                 TAX
750                 PLA
760                 RTI
770                 ;
780                 ;
790                 ;
800 IRQPART2        LDA #BLANK
810                 STA SCREENBLANK
820                 ;
830                 LDA #ON
840                 STA MEGABIT
850                 ;
860                 LDA MOVINGRAST
870                 ASL A
880                 ASL A
890                 ASL A
900                 CLC
910                 ADC #50
920                 STA RASTER
930                 ;
940                 LDA #255
950                 STA COUNT
960                 ;
970                 JSR PRESSF7
980                 ;
990                 JMP NORMIRQ
1000                ;
1010                ;
1020                ;
1030 PRESSF7        LDA CHECKKEY
1040                CMP #FUNCTION7
1050                BNE NOKEYPRESS
1060                ;
1070                LDA SHIFTKEY
1080                AND #SHIFT
1090                BEQ NOSHIFTKEY
1100                ;
1110                LDA MOVINGRAST
1120                BEQ ERROR
1130                ;
1140                DEC MOVINGRAST
1150                ;
1160 ERROR          RTS
1170                ;
1180                ;
1190 NOSHIFTKEY     LDA MOVINGRAST
1200                CMP #24
1210                BEQ ERROR
1220                ;
1230                INC MOVINGRAST
1240                ;
1250 NOKEYPRESS     RTS
1260                ;
1270                ;
1280                ;
1290                ;
1300 MOVINGRAST     .BYTE   0
1310 COUNT          .BYTE   0
1320                ;
1330                ;
1340 END            .END
```

PROGRAM: PROGRAM 1

```
2000 FOR L=0 TO 9:CX=0:FOR D=0 TO 15
   :READ A:CX=CX+A:POKE 49152+L*16+D,A
   :NEXT D
2010 READ A:IF A<>CX THEN PRINT"ERROR
    IN LINE";2040+(L*10):STOP
2020 NEXT L
2040 DATA 120,169,36,141,20,3,169,192,
   141,21,3,169,0,141,14,220,1559
2050 DATA 169,27,141,17,208,169,1,141,
   25,208,141,26,208,169,250,141,2041
2060 DATA 155,192,88,96,169,1,141,25,
   208,238,156,192,173,156,192,208,
   2390
2070 DATA 22,169,27,141,17,208,169,
   252,141,48,208,173,155,192,141,18,
   2081
2080 DATA 208,104,168,104,170,104,64,
   169,123,141,17,208,169,253,141,48,
   2191
2090 DATA 208,169,50,141,18,208,169,
   255,141,156,192,32,97,192,76,49,
   2153
2100 DATA 234,165,197,201,3,208,38,
   173,141,2,41,1,240,16,173,155,1988
2110 DATA 192,201,58,240,8,206,155,
   192,32,142,192,176,248,96,173,155,
   2466
2120 DATA 192,201,250,240,248,238,155,
   192,32,142,192,176,248,96,173,155,
   2930
2130 DATA 192,41,7,201,3,208,2,56,96,
   24,96,0,0,0,255,255,1436
3000 REM ** READY FOR SAVE **
3010 PRINT "[CLEAR,DOWN2,SPC2]DATA
    OK PRESS SPACE TO SAVE"
3020 GET K$:IF K$<>" "THEN 3020
3030 POKE 43,00:POKE 44,192
   :POKE 45,161:POKE 46,192:CLR
3040 SAVE "PROG 1 CODE",8,1
```

PROGRAM: PROGRAM 2

```
2000 FOR L=0 TO 9:CX=0:FOR D=0 TO 15
   :READ A:CX=CX+A:POKE 49152+L*16+D,A
   :NEXT D
2010 READ A:IF A<>CX THEN PRINT"ERROR
    IN LINE";2040+(L*10):STOP
2020 NEXT L
2040 DATA 120,169,36,141,20,3,169,192,
   141,21,3,169,0,141,14,220,1559
2050 DATA 169,27,141,17,208,169,1,141,
   25,208,141,26,208,169,1,141,1792
2060 DATA 142,192,88,96,169,1,141,25,
   208,238,143,192,173,143,192,208,
   2351
2070 DATA 27,72,72,72,104,104,104,169,
   252,141,48,208,169,250,141,18,1951
2080 DATA 208,169,27,141,17,208,104,
   168,104,170,104,64,169,123,141,17,
   1934
2090 DATA 208,169,253,141,48,208,173,
   142,192,10,10,10,24,105,50,141,1884
2100 DATA 18,208,169,255,141,143,192,
   32,109,192,76,49,234,165,197,201,
   2381
2110 DATA 3,208,26,173,141,2,41,1,240,
   9,173,142,192,240,3,206,1800
2120 DATA 142,192,96,173,142,192,201,
   24,240,248,238,142,192,96,0,0,2318
2130 DATA 192,41,7,201,3,208,2,56,96,
   24,96,0,0,0,255,255,1436
3000 REM ** READY TO SAVE **
3010 PRINT "[CLEAR,DOWN2,SPC2]DATA
    OK PRESS SPACE TO SAVE"
3020 GET K$:IF K$<>" " THEN 3020
3040 POKE 43,00:POKE 44,192
   :POKE 45,148:POKE 46,192:CLR
3050 SAVE "PROG 2 CODE",8,1
```

GOING FOR A SONG

Gary Herman brings you the first part of a series which will show you how to make the most of your C64's musical talents.

IT'S A FREQUENTLY REPEATED FACT that Commodore Basic is the worst thing about the 64. From a musical point of view, all the instructions are realised as POKEs (or PEEKs in some instances) to memory locations corresponding to particular registers on the machine's 6581 programmable sound generator.

This makes for a relatively easy transition to machine-code programming — a topic we'll be dealing with in detail later in the series since sound synthesis and music programming are more effective as you get closer to the hardware level. An introduction to Basic techniques is useful because the 64 sound commands are already very close to the hardware level. Unless you use a Basic extension, a high-level language or music utility, music programming in Commodore Basic will inevitably bring you to within spitting distance of the computer's hardware. This is not the case with any other popular machine.

For Basic use, all you really need to know is that location 54272 corresponds to register zero on the 6581 chip and the locations, like the registers, are numbered in steps of one upwards. This is because the 6581 sound interface device is memory-mapped. It is addressed at location D400 HEX (54272 DECIMAL). That is, the chip is enabled when address lines A15, A14, A12 and A10 are all high. The lines A0 to A4 are then used to address SID's registers. While there are 32 possible addresses using A0 to A4, SID actually has only 29 registers. The last three addresses (54301, 54302 and 54303 — or, in hex, D41D, D41E and D41F) are not used.

In general, POKEing a memory location can best be understood as setting certain data lines high and others low. There are three independent sound channels on the Commodore, each one requiring at least five and at most seven different locations to be POKEd. Then there are seven locations which relate to all three channels - three write-only locations and four read-only locations. The write-only locations are the SID registers used to set up the sound you wish to produce. Setting up the sound is just a matter of POKEing the right data into the right location.

Typically, the procedure is first to POKE location 54296 with a volume setting which, as it were, sets up all the channels. This setting is a value between zero (off) and 15 (maximum). Thus 54296 can be divided into two nybbles (four bit numbers), the least significant or right-hand nybble comprising bits number three, two, one and zero of the byte addressed at location 54296. POKEing 54296 with, say, nine sets bit three high (1), bit two low (0), bit one low (0) and bit zero high (1). This gives 1001 as our nybble, in binary code, and 1001 binary is equivalent to nine decimal. The other, high order, nybble at location 54296 is made up of the four most significant bits of the byte — numbers seven, six, five and four. Setting six, five and four high or low has the effect of switching on or off one of the 6581 chip's filter modes. Setting bit seven high or low has the effect of turning off or on the audio output of channel three.

Having set a volume, you must then select the desired channel for output and POKE the two associated locations with

```
100 REM ********************************
110 REM ***LISTING 1********************
120 REM ***ENVELOPE USING BASIC LOOP**
130 REM ***LINEAR DECAY*****************
140 REM ********************************
150 S=54272:L=S:H=S+1:W=S+4:A=S+5:R=S+
6:V=S+24
160 Y=15:F=34:E=240:Q=33:X=10
170 FOR P=0 TO 24:POKE S+P,0:NEXT
180 POKE R,E:POKE H,F:POKE W,Q
190 FOR P=1 TO 40:Y=Y/1.06:POKE V,Y:NE
XT
200 POKE W,0

READY.
```

Program Listing 1

```
READY.

100 REM*********************************
110 REM***LISTING 2*********************
120 REM***NATURAL DECAY*****************
130 REM*********************************
140 S=54272:L=S:H=S+1:W=S+4:A=S+5:R=S+
6:V=S+24
150 Y=30:F=34:E=240:Q=33:X=10
160 FOR P=0 TO 24:POKE S+P,0:NEXT
170 POKE R,E:POKE H,F:POKE W,Q
180 FOR P=1 TO 60:POKEV,Y-3.3*LOG(P):N
EXT
190 POKE W,0
```

Program Listing 2

```
100 REM ********************************
110 REM ***LISTING 3********************
120 REM ***TUNE WITH NOTE DURATION****
130 REM ****VARIABLE T FIXES TEMPO****
140 REM ****A,B&C SET FREQ & DURATION*
150 REM ********************************
160 POKE 54296,15:POKE 54277,9: POKE 5
4278,33
170 READ A,B,C
180 T=20
190 IF A=0 THEN END
200 POKE 54273,A: POKE 54272,B
210 POKE 54276,17
220 FOR P=1 TO T*C:NEXT
230 POKE 54276,0
240 FOR P=1 TO T:NEXT:REM ***DELAY***
250 GOTO 170
260 REM ***DATA:HI FREQ,LO FREQ,DURATI
ON**
270 DATA 19,63,20,21,154,20,17,37,30,8
,147,15,12,216,40,0,0,0

READY.


SEARCHING FOR 4.LISTING4.LISTING
```

Program Listing 3

```
100 REM ***************
110 REM ***LISTING 4***
120 REM ***'K'  SETS***
130 REM *** TUNING  ***
140 REM ***************
150 PRINT CHR$(147);"WAIT":T=100:K=0.5:REM TEMPO AND TUNING
160 DIM F$(3,99),HF(3,99),LF(3,99),D(3,99)
170 DIM N(14),DU(3),WF(3),FW(3),EN(3)
180 FOR X=1 TO 14: READ N(X): NEXT
190 GOSUB 440:POKE 54296,15
200 FOR V=1 TO 3
210 READ AD,SR,HP,LP,FW(V)
220 POKE AD(V),AD:POKE SR(V),SR:POKE PH(V),HP:POKE PL(V),LP
230 NEXT V:I=1
240 FOR V=1 TO 3:READ F$,D(V,I)
250 IF F$="R" THEN F=0:GOTO290
260 IF F$="*" THEN EN(V)=I:O=O+1:GOTO300
270 OC=VAL(RIGHT$(F$,1))
280 F=N(2*(ASC(LEFT$(F$,1))-64)+(LEN(F$)=2)):F=F*2^(OC-4)
290 HF(V,I)=INT(F/256):LF(V,I)=F-HF(V,I)*256
300 NEXT V
310 I=I+1
320 IF O=3 THEN 340
330 GOTO 240
340 FOR K=1 TO I
350 POKE WF(1),0:POKE WF(2),0:POKE WF(3),0
360 FOR V=1 TO 3
370 POKE FH(V),HF(V,K):POKE FL(V),LF(V,K)
380 POKE WF(V),FW(V)
390 NEXT V
400 FOR P=1 TO T:NEXT
410 IF Z=3 THEN END
420 NEXT K
430 END
440 S=54272:FOR X=1 TO 3:CH=7*(X-1)
450 FL(X)=S+CH:FH(X)=FL(X)+1:PL(X)=FH(X)+1:PH(X)=PL(X)+1:WF(X)=PH(X)+1
460 AD(X)=WF(X)+1:SR(X)=AD(X)+1
470 NEXT:RETURN
480 REM***FREQUENCY DATA************************
490 DATA 7381,7818,8271,0,4378,4647,4915
500 DATA 5217,5519,0,5855,6207,6576,6962
510 REM***ENVELOPE,PULSE AND WAVEFORM DATA*****
520 DATA 40,40,7,255,17
530 DATA 9,127,7,255,65
540 DATA 9,255,7,255,17
550 REM***NOTE DATA*****************************
560 DATA A4,8,A4,8,A5,8,C4,12,C4,12,C4,12
570 DATA A4,4,A5,4,A4,4,F3,16,F4,16,F4,16
580 DATA D4,8,D4,8,D3,8,C4,24,C3,24,C4,24
590 DATA A4,8,D2,8,A3,8,L3,12,E3,12,C4,12
600 DATA A4,4,D2,4,A5,4,F4,16,C2,16,F5,16
610 DATA D4,8,D4,8,D5,8,E4,8,E4,8,E5,8
620 DATA C4,8,C4,8,C5,8,B4,8,B3,8,B4,8
630 DATA C4,4,C4,4,C4,4,D4,4,D3,4,D4,4
640 DATA E4,4,G3,4,E4,4,F4,4,F3,4,F4,4
650 DATA G4,4,G3,4,G3,4,A4,4,A3,4,A4,4
660 DATA A#4,8,A#4,8,A#4,8,D4,12,D5,12,G4,12
670 DATA A#5,4,A#4,4,A#3,4,A5,16,A4,16,A3,16
680 DATA G4,8,G3,8,D4,8,F#4,24,F#4,24,D4,24,G4,24,G3,24,D4,24
690 DATA G4,8,G4,8,G3,8,A4,12,C4,12,E4,12,G4,4,E4,16,C4,8
700 DATA C5,8,C4,8,F3,8,C4,8,C4,8,C4,8,F4,4,F4,4,F4,4,A4,4,A4,4,A4,4
710 DATA C5,4,C5,4,C5,4,A#4,4,C5,4,A#4,4,G4,4,E4,4,C4,4
720 DATA A4,8,A4,8,A4,8,C5,12,C4,12,F4,12,A4,4
730 DATA F4,16,F3,16,F4,16,D4,8,D5,8,D4,8,F4,24,F3,24,F4,24
740 DATA C4,24,C3,24,A3,24,D4,8,D4,8,D4,8,F#4,12,D3,12,F#4,12,A4,4,A3,4,A5,4
750 DATA D5,16,D4,16,G4,16,D4,8,A4,8,C5,8
760 DATA B4,24,B3,24,B5,24,B4,8,G4,8,D4,8,B3,4,D4,4,G4,4,B4,4,B4,4,B4,4
770 DATA C5,8,C5,8,C5,8,C4,8,G4,8,C4,8,G4,8,G4,8,C4,8
780 DATA C4,16,E4,16,G4,16,A#4,8,C4,8,E4,8
790 DATA A#3,8,C4,8,C3,8,F4,24,F3,24,F5,24,F4,24,F4,24
800 DATA C4,24,F4,24,A4,24,C4,24,R,24,C3,24,B3,48,B2,48,D4,48
810 DATA A#3,32,E4,32,A#4,32,C4,24,A#4,24,C4,24
820 DATA F4,48,F4,48,A3,24,A#3,24,B3,24,F4,24
830 DATA C4,24,F4,24,C2,24
840 DATA *,0,*,0,*,0
```

Program Listing 4

bytes which set attack, decay, sustain and release characteristics. The ADSR values must be entered before the note is actually turned on, which is a feature of the hardware.

ADSR values are represented by nybbles — attack is the high order nybble of one byte and decay the low order nybble of the same byte, while sustain and release are the high and low order nybbles, respectively, of the byte entered into the next location in memory. Thus there are 16 possible values for each of the ADSR parameters, corresponding to the 16 possible values of one nybble. Decay values, for example, are — in decimal — any number between zero and 15, while attack values (as the high order nybble) are any multiple of 16 between 0*16 and 15*16. They can, of course, be added together to give a combined setting for attack and decay, since we can visualise this process simply as one in which bits are set low or high in both nybbles by determining a value for a single byte. If the value to the POKEs are exact and different powers of two, (0, 1, 2, 4, 8, 16, 32, 64, or 128), the effect of adding them is the same as performing a logical OR on them, since POKEing a location with one of the above powers of two sets precisely one data line high. This point should be borne in mind for future reference.

The table below gives the real values corresponding to ADSR parameter settings. Note that A,D and R are given as rates (that is, the lower their value, the faster that phase of the envelope is over) while S is a proportion of peak volume. During the attack phase, the sound rises to the level set by POKEing 54296 (the peak level). During the decay phase, the sound diminishes to a level set by the sustain value (the plateau level). This can be any of 16 values from the peak value itself (if sustain is set at 15) to zero. Setting sustain to eight, for example, would give a plateau level of roughly half the peak value — that is, for all practical purposes, plateau level = peak value * (sustain value/16). All amplitude values on the 64 increase linearly, which is to say that amplitude (which is heard as a linear increase) you need to double the setting: thus an amplitude of eight is twice an amplitude of four (which is not the case with other computers). Programs 1 and 2 use FOR-NEXT loops to create decaying notes, the first decreasing in volume or amplitude by linear steps, the second by an approximate halving at each step. Notice the difference in effect. Also note the actual locations used.

Value	Attack	Decay/ Release	Sustain
0 (&0)	2 mS	6 mS	0
1 (&1)	8 mS	24 mS	0.07
2 (&2)	16 mS	48 mS	0.14
3 (&3)	24 mS	72 mS	0.20

4 (&4)	38 mS	114 mS	0.27
5 (&5)	56 mS	168 mS	0.34
6 (&6)	68 mS	204 mS	0.41
7 (&7)	80 mS	240 mS	0.48
7 (&8)	100 mS	300 mS	0.54
9 (&9)	250 mS	750 mS	0.60
10 (&A)	500 mS	1.5 S	0.68
11 (&B)	800 mS	2.4 S	0.74
12 (&C)	1 S	3 S	0.80
13 (&D)	3 S	9 S	0.87
14 (&E)	5 S	15 S	0.94
15 (&F)	8 S	24 S	1.00

The sustain figures are only approximate and the timings are based on a 1 MHz clock, while the actual clock rate is 1.02 MHz in the US and 0.98 MHz in Britain. The attack figures give the amount of time taken for the note to rise from zero to whatever the peak amplitude is. Therefore, with a low peak amplitude, the attack will appear gentler than with a high peak amplitude. Similarly, decay and release rates give the amount of time taken for the note to decay or release to zero amplitude. If a non-zero sustain value is set, the decay will be interrupted before the specified time is up. If a note decays from a low peak value, or is released from a low plateau value, then the time taken is as specified, so that the decay or release will be gentler than if the peak or plateau values were higher.

Having set overall amplitude and envelope parameters, the next step is to set frequency. The Commodore has a range of almost eight octaves. The nominal frequencies can be calculated using the formula: F=N*C/16777216 Hz, where N is the decimal equivalent of the familiar two byte (16 bit) frequency number and C is the clock rate.

Data is entered as a two byte number (that is, as two bytes in consecutive locations and is, again, linearly related to frequency rather than pitch. Pitch is the term we use to describe the sensation of music at frequency, in which the basic units are notes and tones. Two notes, for example C and D, will have a much greater difference in frequency at a high pitch than at a low pitch, which means that errors may creep in at the bottom end of the 64's octave scale, where small numerical differences can have a large effect on pitch.

Since the resolution is so good — 65536 values covering a range of 90-plug notes — accurate tuning is easy, if somewhat tedious. The most accurate, if least efficient, method for entering frequencies is to specify the frequency data for each note to be used in a program as one of two items in a DATA statement. (The relationship between the two byte representation of frequency and a single number value is given by FN=256*HB+LB, where FN is a frequency number and HB and LB are the equivalent high and low bytes in decimal). This method makes overall tuning difficult and it is often the best compromise to actually calculate values from a core of one octave's worth of data in your program, including a tuning variable in the calculation. Program 3 plays a tune using frequencies entered as DATA, while Program 4 uses core data to calculate frequencies. Note how easy it is to tune Program 4 by just changing a single parameter in the calculation.

```
100 REM ********************************
110 REM *         LISTING 5            *
120 REM *    WAVEFORM  ADDITION        *
130 REM *                              *
140 REM *PULSE  WIDTH VARIATION SHOWN*
150 REM *                              *
160 REM *BITS  SET ON W/FORM REGISTER*
170 REM *                              *
180 REM *INPUT AND COMBINE  WAVEFORMS*
190 REM ********************************
200 POKE 54296,15
210 POKE 54272,75:POKE 54273,34:POKE 5
4274,255
220 POKE 54277,0:POKE 54278,240
230 PRINT"[CLEAR]"
240 PRINT "[HOME][DOWN][DOWN][DOWN][DO
WN][DOWN][DOWN][DOWN][DOWN][DOWN][DOWN
][DO
WN][DOWN][DOWN][DOWN][DOWN][DOWN][DOWN
][DOWN][DOWN][DOWN][DOWN]ENTER 2 WAVEF
ORMS
(0 OR 1=T,2=S,3=P,4=N)"
250 INPUT A,B
260 IF A<0 OR A>4 OR B<0 OR B>4 THEN P
RINT"[UP][UP][UP][DOWN][LEFT][LEFT]":G
OTO2
40
270 POKE 54276,8:POKE 54276,0
280 A=-16*(A>0)*2^(A-1):B=-16*(B>0)*2^
(B-1)
290 K=0
300 IF A=64 OR B=64 THEN K=1
310 POKE 54276,(AORB)+1
320 PRINT "[CLEAR]";TAB(240);"WAVEFORM
 NO:";A;"+";B;"+";1;"=";(AORB)+1;
330 IFPL=1THENFORPW=0TO15:POKE54275,PW
:PRINT"[HOME]";TAB(255);"PW";INT((PW+1
)*10
0/16);"%":NEXT
340 B7=(A=128)OR(B=128)
350 B6=(INT((INT((AORB)/64))/2)<>(INT(
(AORB)/64)/2))
360 B5=(INT((INT((AORB)/32))/2)<>(INT(
(AORB)/32)/2))
370 B4=(INT((AORB)/32)<>(((AORB)/32)))
380 PRINT"[DOWN][DOWN][DOWN][DOWN][DOW
N][DOWN][DOWN][DOWN][DOWN]"
390 PRINT "WAVEFORM REG: BIT 7.BIT 6.B
IT 5.BIT 4"
400 PRINT "                 --------------
------------"
410 PRINT"                 [RVSON]   ";AB
S(B7);"   ";ABS(B6);"   ";ABS(B5);"
";AB
S(B4)

420 PRINT "                 --------------
------------"
430 IF K=0 THEN 460
440 FORW=0TO15:POKE54275,W:PRINT"[HOME
]";"PW";INT((W+1)*100/16);"%"
450 FOR P=1 TO 100:NEXTP,W
460 PRINT"[HOME]";TAB(130);"HIT A KEY
TO GO AGAIN"
470 GETA$:IFA$=""THEN470
480 GOTO240
```

Program Listing 5

Frequency data is entered as a low byte and a high byte in consecutive locations — the low byte allowing fine tuning, the high byte allowing coarse sweeping through octaves. The range of values is, of course, zero to 255 for each byte and out-of-range values will — in this and all other cases — either return an error and stop the program or interfere with other parameters (in the case, for example, when you POKE the volume register with a value greater than 15 you will interfere with the filtering parameters).

After frequency comes waveform. There are four 'pure' waveforms: triangle, sawtooth, pulse and white noise. They can be 'mixed' but the effects are unpredictable, since the waveforms are not simply added but logically ANDed. Thus 16, 32 and 64 produce, respectively, triangle, sawtooth and pulse waves when POKEd to the relevant location. Adding 16 and 32 (or performing a logical OR on the values) produces a very thin pulse as a result of logically ANDing the waves, but adding 16 and 65 actually produces a fairly rich composite waveform with a shape rather like the top half of a sine wave (Program 5).

If pulse is chosen, as in the above example, it is necessary to set a pulse-width or duty cycle. There are 4096 possible values set by POKEing a byte and a nybble into consecutive locations. The byte has a range of zero to 255 and the nybble a range of zero to 15. The reason for setting different pulse widths is that pulse width does affect timbre, because the harmonic content of a pulse wave at a

given frequency depends on duty cycle. A value of zero POKEd into both registers or a value of 4095 (that is, 15 in the high nybble and 255 in the low byte) wil give a constant DC output. A value of 2048 (eight in the high nybble and zero in the low byte) will give a square wave. It is worth experimenting with combinations of waveforms, as above, using different width pulses plus triangle or sawtooth waves, since the logical ANDing involved works on the harmonics, cancelling out some and magnifying others.

Setting the waveform should be the last POKE in any series of commands, because the waveform registers are also the control registers for any channel. The values for waveforms are as given above plus 128 for noise, but the sound will only be heard if bit zero of the control register is set to one. This is known as the gate bit, and it triggers the start of the attack phase of any sound. If it is set to zero, the sustain phase of the note terminates and it enters its release phase. Thus, to start a note playing, for instance a sawtooth wave, the location corresponding to the relevant waveform/control register must be POKEd with 33. To turn the note off, POKE the same location with zero on any even number. Thus a note duration is governed by the attack time (given in the table above), plus the proportion of the decay time it takes to reach the sustain level (given by decay time * (1 — sustain level), where decay time and sustain level are the figures given in the table above for the relevant values POKEd into their assocaited locations), plus the duration of the sustain phase, plus the release time (given above). Sustain duration is set by means of a delay in the program. This can be almost any command, but typically, a FOR. . .NEXT loop is used. Other common delay techniques use the 64's internal clock to measure a fixes amount of time (the variable TI holds clock data), a WAIT command to detect a particular event or a GET command to detect the presing of a key. Problems sometimes occur with the 64's keyboard buffer, which should be cleared by a POKE 198,0 instruction if a key is pressed to initiate a note.

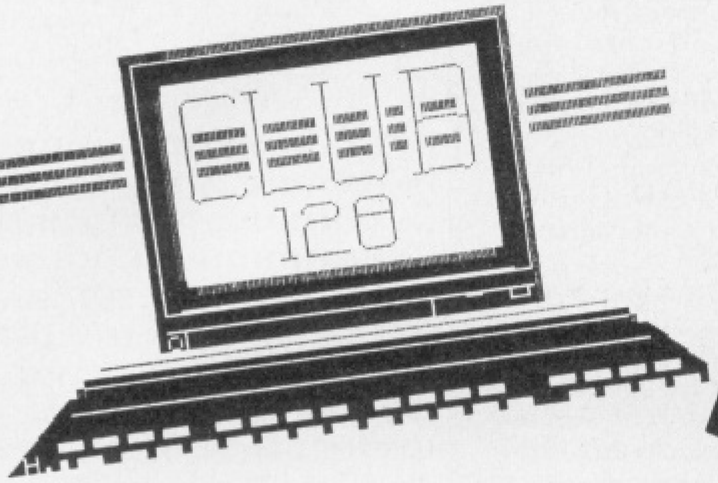
The remaining sound locations will be dealt with in a future episode of the series, after which we'll move on to machine-code and a discussion of some theoretical aspects of music-making. Meanwhile, you should note that all locations and data used in the sample programs have been ascribed to variables to simplify the actual POKEs. This not only makes the program easier to understand, but it also saves time, since the 64 deals with variables quicker than it does with numeric constants. To finish, Program 6 shows what can be done with the 64's filter facility as a taster of things to come.

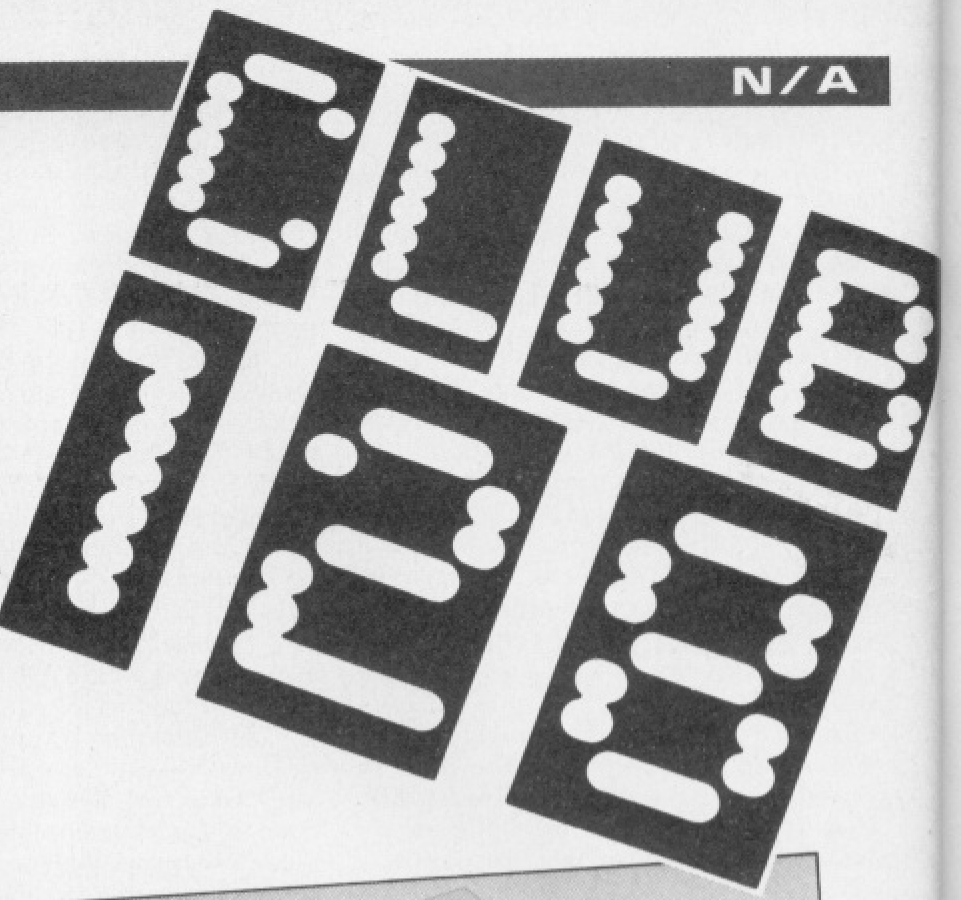
```
SEARCHING FOR 6.LISTING6.LISTIN

READY.


100 REM***********************************
110 REM*********  LISTING 6  **********
120 REM*********FILTER EFFECT**********
130 REM***********************************
140 POKE 54296,31:POKE 54277,64: POKE
54278,128:POKE 54293,0:PRINT CHR$(147)
150 FOR R=0 TO 240 STEP 240:POKE 54295
,1+R
160 PRINT CHR$(19);TAB(10);"RESONANCE"
;R
170 READ A,B
180 IF A=0 THEN 230
190 POKE 54273,A: POKE 54272,B
200 POKE 54276,33
210 FOR F=0 TO 255 STEP 2:POKE 54294,F
:NEXT
220 GOTO 170
230 FOR P=1 TO 800: NEXT: POKE 54276,0
240 RESTORE:NEXT
250 DATA 19,63,21,154,17,37,8,147,12,2
16,0,0
```

Program Listing 6





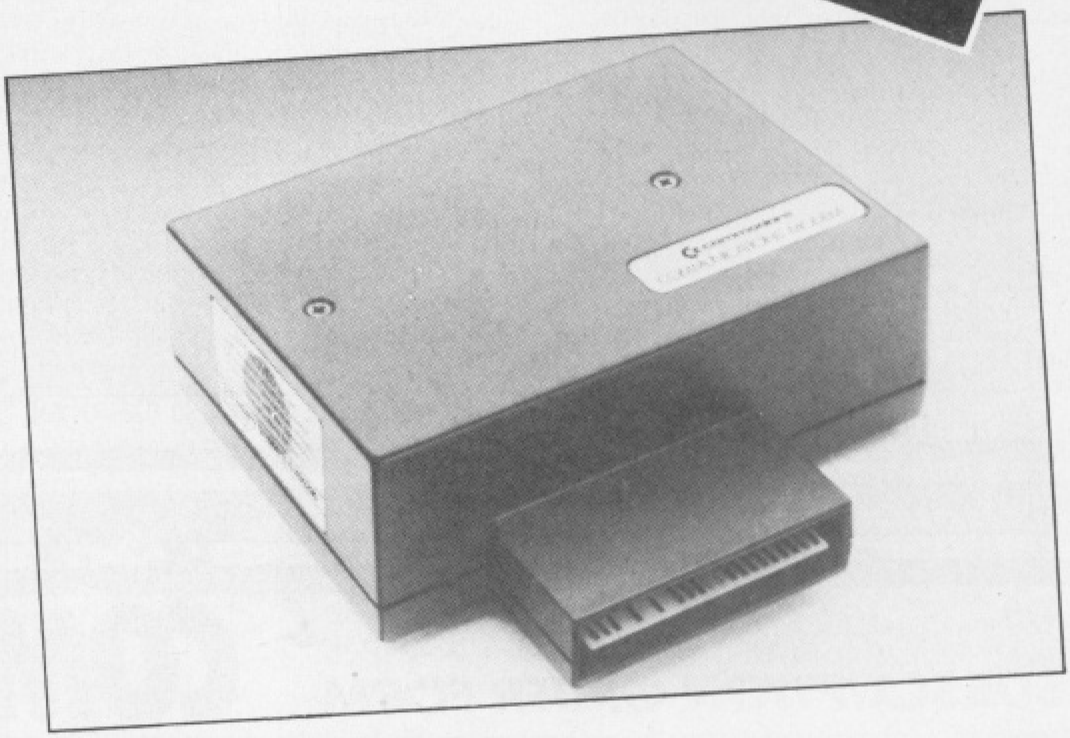
HAVE YOU EVER WANTED TO GET INTO communications? Would you like to get in touch with other Commodore owners? Well, Your Commodore, together with Compunet, are pleased to announce the start of Club 128, and give you a chance of a special membership offer.

Club 128 will appear on Compunet as a 'free access' area open to all CNET subscribers.

The club will be the focal point for all 'serious' users of Commodore computers, hence the '128' affiliation. The 128 doesn't mean that owners of C64s can't join. In fact the club is open to all Compunet members though only 'serious' users are expected to make a great deal of use of it.

Many sections of Compunet are due to be 'pulled in' to this online area, including the 'Briefcase' business section that is already in existence.

Your Commodore will have a magazine section within the club area where you will be able to find news and the latest programs that are published in the Magazine.



The Offer

So what do you get and what does it cost?

Your Commodore is offering a special membership to the Club for £69.99 (inc VAT and P&P in the UK) including:

A Commodore 64/C128 Communications modem (worth £79.99)

A three month GOLD subscription to Compunet (worth £15.00)

A six month subscription to Your Commodore

*** PLUS ***

£10.00 of discounts on selected business and communications software, including software for accessing Viewdata and Electronic Mail services such as Prestel and Telecom Gold.

In order to take advantage of this special offer you simply need to send a cheque or Postal Order made payable to Club 128 to the address on the below order form.

I enclose a cheque made payable to Club 128 for £69.99 for my Gold membership to Compunet, Commodore modem and six month subscription to Your Commodore.

Name ....

Address ....

....

....

Postcode ....

Please allow at least 28 days for delivery. Order should be sent to: Club 128 Compunet Teleservices Ltd, 7-11 Minerva Rd, London NW10 6HJ.



YOU AND THE ENEMY — ONLY ONE OF YOU WILL SURVIVE
Your AWAT (All Weather-All Terrain) jet stands fully fuelled and fully armed. A vast invasion fleet is anchored just off your shores. Ground forces have come ashore and are advancing on your positions protected by massive air cover.
You are the last fighter pilot. Your country turns to you and asks ... "Are you good enough to be called an ACE?"
C64 VERSION — ZZAP! 64
'FLIGHT SIMULATOR OF THE YEAR'
OUT NOW —
SPECTRUM VERSION £9.95
C PLUS/4 DISK VERSION £12.95
(FULL 64K PLUS/4 VERSION)
THE · ULTIMATE · COMBAT · JET · SIMULATOR
ACE
FREE WITH EVERY COPY POSTER
MISSILE AVOIDED
COLLISION WARNING
LOW ALTITUDE
Featuring tanks, helicopters, hills, trees, ships and a totally unique refuelling sequence ACE is the fastest, most exciting air combat game available. With twin flying mode (using two joysticks) you can fly and fight with your friend as weapons man.
ACE — THERE IS NO OTHER GAME LIKE IT!
cascade
CASCADE GAMES LTD
Cascade Games Ltd., Harrogate, HG1 5BG, England. Telephone: (0423) 504663
Available from all good software suppliers or direct from Cascade - 24 hour ordering service on (0423) 504663
Versions available for: SPECTRUM 48K COMMODORE C 64/128, C 16, VIC20, PLUS/4
CASSETTE £9.95
DISK £12.95 (C 64/128, Plus/4 only)
Screen shots are taken from the C 64 version; C 16 and VIC20 do not have ground objects.

Save your fingers a lot of work with our new software service.

SOFTWARE FOR SALE

IT'S THREE O'CLOCK IN THE MORNING. You sit at the computer keyboard having just finished a marathon typing session entering one of the superb programs from Your Commodore. Your fingers reach for the keyboard and press the letters R, U and N/. You sit back expectantly and...nothing happens.

Well, I'm sure that we have all had problem before now. When it does happen it's a matter of spending hours searching through the program for any typing mistakes. No matter how long you look or how many people help you, you can usually guarantee that at least one little bug slips through unnoticed.

Here, at Your Commodore, we pride ourselves on the quality of listing that we print. Unfortunately, this usually means that they are also very long, thus taking longer to type in and leaving more room for errors. All of the listings in Your Commodore are taken straight from a printout of working programs, it is therefore very unusual for errors to appear in the magazine.

Because of the length of our programs we do get a large number of requests from readers who would like us to put specific programs on tape or disk for them. Obviously this is very time consuming and means that we can't spend as much time working on the magazine as we would like.

We are therefore proud to announce the start of the 'Your Commodore Software Service'. Most of the programs from each issue of the magazine will now be available on a single cassette for a price of just £4.00. We will not be making disks available since they would have to be a lot more expensive and more difficult to post. This shouldn't cause you any problems though as none of the programs will be protected and it will be a simple matter to save the programs to disk yourselves.

All programs on the cassette will be saved using a tape turbo routine. However, we cannot guarantee that all programs will work correctly with this turbo routine present. We therefore recommend that before you use any of the programs you make a copy of the programs on your own cassette or disk and use this version of the program **not** the original.

PLEASE COMPLETE IN BLOCK CAPITALS **ORDER FORM**

All cheques/postal orders should be made payable to:-
ARGUS SPECIALIST PUBLICATIONS LTD

I would like to order the following cassettes: (please tick box)

APR '86	**YCAPR86**	**£4.00**	
MAY '86	**YCMAY86**	**£4.00**	
JUNE '86	**YCJUN86**	**£4.00**	
JULY '86	**YCJUL86**	**£4.00**	

I enclose a cheque/postal order for £ ....

NAME..

ADDRESS..

....

....

POSTCODE..

All orders should be sent to:-
YOUR COMMODORE SOFTWARE, READERS SERVICES, ARGUS SPECIALIST PUBLICATIONS,
WOLSEY HOUSE, WOLSEY ROAD, HEMEL HEMPSTEAD, HERTFORDSHIRE HP2 4SS.

Please allow at least 28 days for delivery.

LUNAR ORDEAL

John Fletcher proves that good games are possible in Basic.

YOUR CITY IS UNDER attack from the alien 'Honker'. Your mission is to rescue all of your people from the city before 'Honker' reaches it. Should he reach it, then you can wave good bye to your three lives.

There may seem to be nothing original about the game but there are a couple of things that make it worth looking at. Firstly it is fun to play presented. Secondly, John Fletcher has proved that not all good games need to be written totally in machine code, Lunar Ordeal is written mainly in Basic.

A joystick in port 2 is needed to play the game.

Variables

Scoring Variables and strings:
Sc,SC$,S1$,CA,N$,C,SN$,Q$,W
Module Movement:
W,H,VV,V1,H1,VO,HO,U,D
Miscellaneous:
L — Lives
LV — Level
KY — Joystick Movement
FU — Fuel
XX — Print title page if XX=1

PROGRAM: LUNAR ORDEAL

```
0 DIM SN$(10),SC(10)
  :FOR T=1 TO 10
  :SN$(T)="NO ONE YET !!"
  :SC(T)=500-(20*T):NEXT
1 IF PEEK(49152)<>120 THEN P
  OKE 53281,1:POKE 53280,1
  :GOSUB 16000:
2 SYS 49152:XX=1:L=3:LV=1
  :U=-.5:D=0:GOSUB 16100
  :GOSUB 30:XX=0
  :SC$="000000"
3 PRINT CHR$(8):POKE 56325,72
  :GOSUB 10000:CA=0
  :SE$="000000":POKE V+21,
  255
10 PRINT"[CLEAR]":FU=20
11 PRINT"[HOME]"TAB(13)"
  [GREEN]FUEL :[RVSON,
  YELLOW]))))))))))))))))))
  ))"
12 PRINT"[HOME,WHITE]LEVEL";
  LV
13 PRINT"[HOME,DOWN,C6]
  ****************************
  ***************";
20 POKE 53281,0:POKE 53280,0
  :POKE V+31,0
25 Y=1904:YY=56176
  :POKE 2040,240
30 V=53248:POKE V+1,63
  :POKE V,155:POKE V+21,63
  :POKE V+45,2:POKE V+39,1
  :POKE V+41,11
31 POKE V+28,255:POKE V+37,2
  :POKE V+38,7
40 POKE V+2,0:POKE V+3,80
  :POKE V+4,0:POKE V+5,120
  :POKE V+6,0:POKE V+7,160
  :POKE V+8,0
41 POKE V+9,190:POKE V+10,10
  :POKE V+11,213
42 POKE V+40,9
50 POKE V+27,255:POKE V+23,6
  :POKE V+29,6:POKE V+44,6
  :P=56156
55 IF XX=1 THEN RETURN
56 POKE 53272,(PEEK(53272)AN
  D 240)+12:POKE V+17,
  PEEK(V+17)OR 64
60 FOR I=0 TO 39:POKE 56216+
  I,8:POKE 1944+I,43:NEXT
  :PRINT"[HOME]"
61 FOR I=0 TO 39:POKE 56256+
  I,8:POKE 1984+I,44:NEXT
69 PRINT TAB(17)"[DOWN19,
  YELLOW])[SPC5])"
70 PRINT TAB(17)"[YELLOW])
  [C8]'[BLUE]***[C8](
  [YELLOW])"
71 PRINT TAB(17)"[YELLOW])
  [C8]%,,,&[YELLOW])"
72 PRINT TAB(17)"[RVSON,
  YELLOW])[C8,RVSOFF](,,,'
  [RVSON,YELLOW])[RVSOFF]"
75 POKE Y+16,45:POKE YY+16,1
  :POKE Y+24,45:POKE YY+24,1
  :POKE V+30,0:POKE V+31,0
76 PRINT"[RVSON,WHITE]SCORE
  :"SC$;:PRINT"[HOME]"
90 FU=20:M=150:SYS 49152
  :VV=65:H=155:VO=0:HO=0
  :POKE 54296,15
  :POKE 54278,240
100 V1=.1:H1=0
104 :
105 REM ******** MAIN LOOP
  *********
106 :
110 KY=PEEK(56320)
111 POKE 1043+FU,32
  :POKE P,INT(RND(1)*16)
115 IF RND(1)>.7 THEN POKE Y
  +16,45:POKE Y+24,45
120 IF KY=126 AND FU>0 THEN
  V1=U:FU=FU-.5:POKE 54273,8
  :POKE 54276,129
  :POKE V+21,127
125 IF FU=0 THEN PRINT"[HOME]
  "TAB(22)"[RVSON,WHITE]OUT
  OF FUEL"
130 IF KY=123 THEN H1=-.2
140 IF KY=119 THEN H1=.2
150 IF KY=127 THEN V1=.1:H1=0
  :POKE 54276,0
160 IF H1=H9 THEN GOTO 180
170 H9=H1:K=SGN(ABS(H9))*129
  :POKE 54273,39
  :POKE 54276,K
180 VO=VO+V1:HO=HO+H1
200 IF VV<50 THEN VO=ABS(VO)
210 IF H<20 THEN HO=ABS(HO)
220 IF H>240 THEN HO=-ABS(HO)
250 VV=VV+VO:H=H+HO
260 POKE V,H:POKE V+1,VV
  :POKE V+12,H:POKE V+13,
  VV+14
320 Z=Z+.5:IF Z=1 THEN POKE
  2045,243
321 IF Z=2 THEN POKE 2045,242
  :Z=0
330 IF INT(VV)=200 AND INT(H
  )>172 AND INT(H)<177 THEN
  GOTO 1000
340 IF PEEK(V+31)=1 THEN GOT
  O 2002
350 IF PEEK(V+30)>2 AND PEEK
  (V+30)<8 THEN GOTO 2000
360 IF PEEK(V+10)=140 THEN G
  OSUB 3000
500 IF RND(1)>.7 THEN POKE Y
  +24,46:POKE Y+16,46
510 POKE V+21,63
520 GOTO 100
997 :
998 REM ******* LANDING ROUT
  INE ********
999 :
1000 FOR T=690 TO 715
  :POKE T,0:NEXT
  :POKE 54276,0:POKE 54273,
  30:
1001 PRINT"[HOME,DOWN2]"
  :A$="[C4,C5,C8,WHITE]"
  :FOR A=1 TO 8:FOR B=1 TO 4
  :PRINT TAB(17)MID$(A$,B,
  1);
1002 PRINT"[UP]LANDED!"
1003 FOR T=1 TO 60:NEXT T,B,A
1004 POKE V+31,0:POKE V+23,
  128:POKE V÷29,128
  :POKE 54278,3
1005 FOR I=1 TO FU*3:CA=CA+I
  :S$=STR$(CA):S1$=RIGHT$(S
  $,LEN(S$)-1):POKE 54276,32
1006 SC$=LEFT$(SE$,
  6-(LEN(S1$)))+S1$
  :POKE 1043+FU,32:FU=FU-.3
1007 POKE 646,I:PRINT"[HOME,
  DOWN24]"TAB(6)SC$;
1008 POKE 54276,17:NEXT I
  :POKE 54296,0:POKE 1043,32
  :POKE 1044,16
1010 POKE Y+16,32
  :POKE Y+17,46:FOR T=1 TO
  150:NEXT
1015 POKE Y+17,41
```

Program Breakdown

0 — Variables for high score table
1 — Check if machine code is in memory
2-3 — Go to title page routine
10-90 — Print screen, set up sprites and variables
110-520 — Main loop for game
1000-1008 — Routine for landing and scoring
1010-1041 — Animate people going into lifts
1059-1090 — Bring mother ship down to collect module
2000-2050 — Routine for hitting something, deduct life
3000-3030 — Honker has reached people, end of game
5000-5030 — Game over, check for high score
5040-6050 — Input name for high score, calculate place, print names
1000-1068 — Title page
16000-16111 — Read data into appropriate memory locations, set sprite speeds and directions.
17000-17040 — Data for machine code
20000-20048 — Data for sprites
20060-20065 — Data for user defined graphics.



```
   :POKE Y-23,46:FOR T=1 TO
   150:NEXT
1020 POKE Y-23,41
   :POKE Y-63,46:FOR T=1 TO
   150:NEXT
1021 POKE Y-63,41
   :FOR T=1 TO 150:NEXT
1030 POKE Y+24,32
   :POKE Y+23,46:FOR T=1 TO
   150:NEXT
1035 POKE Y+23,41
   :POKE Y-17,46:FOR T=1 TO
   150:NEXT
1040 POKE Y-17,41
   :POKE Y-57,46:FOR T=1 TO
   150:NEXT
1041 POKE Y-57,41
   :FOR T=1 TO 150:NEXT
1059 POKE V+21,129
   :POKE V+45,1
1060 POKE V+14,162
   :POKE V+15,0
1070 FOR I=0 TO 220
   :POKE V+37,I:POKE V+15,I
   :NEXT:POKE V,174
1080 FOR I=220 TO 17 STEP-1
   :POKE V+37,I:POKE V+15,I
   :POKE V+1,I-16:NEXT
1090 LV=LV+1:POKE 56325,
   50-(LV*1.5):GOSUB 16100
   :GOTO 10
1997 :
1998 REM******** CRASHED ROUT
   INE ********
1999 :
2000 IF PEEK(V+30)=65 THEN P
   OKE V+30,0:GOTO 500
2001 IF PEEK(V+30)=0 THEN GO
   TO 500
2002 FOR T=690 TO 715
   :POKE T,0:NEXT
   :POKE 54276,0:POKE V+38,2
   :POKE V+39,7
2003 PRINT"[HOME,DOWN2]"
   :A$="[C4,C5,C8,WHITE]"
   :FOR A=1 TO 8:FOR B=1 TO 4
   :PRINT TAB(17)MID$(A$,B,
   1);
2005 PRINT"[HOME]"TAB(17)"
   [DOWN2]CRASHED!"
2006 FOR T=1 TO 70:NEXT:NEXT
   :NEXT:L=L-1
2010 POKE 54276,128
   :POKE 54277,59
   :POKE 54278,14
   :POKE 54273,3:POKE 54272,2
2015 POKE 54276,129
   :POKE 2040,246
   :FOR T=1 TO 200:NEXT T
2039 PRINT TAB(17)"[DOWN,C7]
   LIVES:";L
2040 IF L=<0 THEN 5000
2050 FOR T=1 TO 1000:NEXT
   :GOSUB 16100:POKE V+30,0
   :POKE V+31,0:GOTO 10
2997 :
2998 REM***** HONKER HAS MAD
   E IT *****
2999 :
3000 FOR T=690 TO 715
   :POKE T,0:NEXT
   :POKE 54276,0:FOR D=1 TO 3
3001 PRINT"[HOME,DOWN3]"
   :A$="[BLACK,C4,C5,C8,
   WHITE]":FOR A=1 TO 4
   :FOR B=1 TO 5:PRINT MID$(
   A$,B,1);
3002 PRINT"[UP]OH NO! HONKER
   HAS MADE IT TO YOUR PEOP
   LE";
3003 FOR T=1 TO 60
   :NEXT T,B,A,D:L=L-1
   :IF L=<0 THEN 5000
3010 PRINT TAB(15)"[YELLOW]
   ONE LIFE LOST. LIVES :";L
3020 FOR T=1 TO 2000:NEXT
3030 GOTO 10
4997 :
4998 REM******* GAME OVER
   *******
4999 :
5000 FOR T=690 TO 715
   :POKE T,0:NEXT
   :FOR T=1 TO 90
   :POKE 54276,0:POKE 54273,
   30
5001 POKE 646,INT(RND(1)*16)
   :PRINT"[HOME]"TAB(16)"
   [DOWN2]GAME OVER!":NEXT T
5004 POKE V+31,0:POKE V+21,0
5010 PRINT"[CLEAR,DOWN,WHITE]
   YOUR FINAL SCORE WAS";CA
5020 PRINT"[DOWN]AND MANAGED
   TO GET TO LEVEL ";LV
   :FOR T=1 TO 5000:NEXT
   :POKE 198,0
5030 IF CA<SC(10)OR CA=0 THE
   N 5080
5040 OPEN 1,0
5041 POKE 53281,3
   :POKE 53280,3:PRINT"
   [CLEAR,DOWN7,RIGHT7,BLACK]
   TYPE IN YOUR NAME PLEASE"
5045 PRINT"[DOWN,RIGHT9,SPC]
   13 LETTERS MAXIMUM "
5046 PRINT TAB(12)"[BLUE,
   DOWN3].............
   [LEFT13]";
5047 INPUT#1,N$:IF LEN(N$)>1
   4 THEN CLOSE 1:GOTO 5040
5050 SN$(10)=N$:SC(10)=CA:
5055 C=0:FOR T=1 TO 9
5060 IF SC(T+1)>SC(T)THEN W=
   SC(T+1):SC(T+1)=SC(T)
   :SC(T)=W:GOTO 5076
5070 NEXT:IF C=1 THEN 5055
5075 GOTO 5080
5076 Q$=SN$(T+1):SN$(T+1)=SN
   $(T):SN$(T)=Q$:C=1
   :GOTO 5070
5080 POKE 53280,0
   :POKE 53281,0:POKE 53272,
   28
5091 PRINT"[CLEAR,DOWN,
   RIGHT2,WHITE]-[YELLOW,SPC]
   FREEKY PLAYERS ALL TIME
   BEST[SPC,WHITE]-"
5092 PRINT"[GREEN]
   **************************
   ***********"
5095 FOR T=1 TO 10:POKE 646,T
   :PRINT"[DOWN,RIGHT5]"T;
   SN$(T)TAB(25)SC(T):NEXT T
6000 PRINT"[HOME]"TAB(4)"
   [DOWN24,WHITE]PRESS ANY
   KEY TO PLAY AGAIN"
6001 CLOSE 1:H=0:DE=0
6010 TT=PEEK(197)
```

```
   :IF TT=64 THEN DE=DE+1
6015 IF DE=400 THEN DE=0
   :GOTO 2
6020 IF TT<>64 THEN GOTO 2
6050 GOTO 6010
9997 :
9998 REM ####### TITLE PAGE
   #######
9999 :
10000 POKE 53281,1
   :POKE 53280,1:PRINT"
   [CLEAR,C3]":POKE 2040,240
   :POKE V+39,14
10001 POKE 53272,21
   :POKE V+17,155:POKE V+37,5
   :POKE V+38,13:CA=0
10010 PRINT TAB(2)"[UP,SU,
   SC2,SI,SPC15,SU,SC4,SI]
10011 PRINT TAB(2)"[SB,RVSON,
   SPC2,RVSOFF,SB,SPC15,SB,
   RVSON,SPC4,RVSOFF,SB]
10012 PRINT TAB(2)"[SB,RVSON,
   SPC2,RVSOFF,SB,SPC15,SB,
   RVSON,SPC,RVSOFF,SU,SI,
   RVSON,SPC,RVSOFF,SB]
10013 PRINT TAB(2)"[SB,RVSON,
   SPC2,RVSOFF,SB,SPC15,SB,
   RVSON,SPC,RVSOFF,SB2,
   RVSON,SPC,RVSOFF,SB]
10014 PRINT TAB(2)"[SB,RVSON,
   SPC2,RVSOFF,SJ,SC3,SI,
   SPC11,SB,RVSON,SPC,RVSOFF,
   SB2,RVSON,SPC,RVSOFF,SB]
10015 PRINT TAB(2)"[SB,RVSON,
   SPC6,RVSOFF,SB,SPC11,SB,
   RVSON,SPC,RVSOFF,SJ,SK,
   RVSON,SPC,RVSOFF,SB]
10016 PRINT TAB(2)"[SB,RVSON,
   SPC6,RVSOFF,SB,SPC]U N
    A R[SPC3,SB,RVSON,SPC4,
   RVSOFF,SB,SPC]R D E A L
10017 PRINT TAB(2)"[SJ,SC6,
   SK,SPC11,SJ,SC4,SK]
10020 PRINT TAB(3)"[DOWN,C2]
   BY JOHN FLETCHER[SPC2](C)
    1985 AGE 15[SPC,BLUE]"
10021 PRINT"[DOWN]
   TRY TO LAND ON THE PAD
    AT THE BOTTOM OF"
10022 PRINT"THE SCREEN BEFOR
   E '[RED]HONKER[BLUE]
   ' ATTACKS YOUR"
10023 PRINT"PEOPLE .SHOULD
   HE ACHEIVE THIS THEN YOU"
10024 PRINT TAB(2)"WILL LOSE
   ALL THREE OF YOUR LIVES."
10025 PRINT"[DOWN,SPC]CONTROL
   :- JOYSTCK IN PORT TWO"
10027 PRINT" UP[SPC3]= MAIN
   [SPC2]THRUST (USES FUEL)"
10028 PRINT" LEFT = LEFT
   [SPC2]THRUST[SPC5]"
10029 PRINT" RIGHT= RIGHT
   THRUST[SPC5]"
10055 PRINT TAB(9)"[BLACK,
   DOWN]PRESS ANY KEY TO PLA
   Y"
10058 PRINT"[SPC,DOWN,BLUE]
   I'M[SPC,RED]'HONKER'"
10060 TT=PEEK(197)
   :IF TT=64 THEN DE=DE+1
10065 IF DE=700 THEN DE=0
   :POKE V+21,0:GOTO 5080
10066 IF TT<>64 THEN RETURN
10067 A1=A1+1:IF A1=15 THEN
   POKE 2045,243
10068 :IF A1=30 THEN POKE 20
   45,242:A1=0
10100 GOTO 10060
15000 :
15001 REM ##### INTIALISATIO
   N[SPC2]#######
15002 :
16000 PRINT"[CLEAR]"TAB(12)"
   [DOWN11,BLACK]PLEASE WAIT
   !"
16002 FOR I=0 TO 90:READ A
   :POKE 49152+I,A:NEXT
   :SYS 49152
16010 FOR I=2041 TO 2045
   :POKE I,241:NEXT
   :POKE 2040,240
   :POKE 2045,242
   :POKE 2046,244
16020 POKE 2047,245
16050 FOR K=0 TO 6
   :FOR J=0 TO 63:READ A
   :POKE(240+K)*64+J,A
   :NEXT J,K
16051 POKE 56334,
   PEEK(56334)AND 254
   :POKE 1,PEEK(1)AND 251
16052 FOR I=0 TO 511
   :POKE 12288+I,
   PEEK(53248+I):NEXT
16060 FOR I=0 TO 95:READ A
   :POKE 12568+I,A:NEXT
16061 POKE 1,PEEK(1)OR 4
   :POKE 56334,PEEK(56334)OR
   1
16100 POKE 690,2:POKE 691,0
   :POKE 692,2:POKE 693,0
   :POKE 694,3:POKE 695,0
   :POKE 696,1
16101 POKE 697,0:POKE 698,20
   :POKE 699,0
16110 POKE 706,255:POKE 707,0
   :POKE 708,1:POKE 709,0
   :POKE 710,1:POKE 711,0
16111 POKE 712,1:POKE 713,0
   :POKE 714,1:POKE 715,0
16299 RETURN
16997 :
16998 REM ##### MACHINE CODE
   DATA #####
16999 :
17000 DATA 120,169,192,141,
   21,3,169,13,141,20,3,88,
   96,162,15,169,128
17010 DATA 141,61,3,141,60,3,
   222,208,2,208,44,189,176,
   2,157,208,2
17020 DATA 189,192,2,240,33,
   16,12,189,0,208,8,222,0,
   208,40,240,7
17030 DATA 208,19,254,0,208,
   208,14,173,60,3,208,9,173,
   61,3,77,16,208
17040 DATA 141,16,208,173,60,
   3,208,3,78,61,3,73,128,
   141,60,3,202,16,191,76,49,
   234
19997 :
19998 REM ####### SPRITE DAT
   A #######
19999 :
20000 DATA 0,0,0,0,40,0,0,
   150,0,2,85,128,9,85,96,9,
   125,96,37
20001 DATA 255,88,39,255,216,
   149,125,86,149,85,86,37,
   85,88,10,170
20002 DATA 160,0,130,0,2,0,
   128,8,0,32,32,0,8,32,0,8,
   168,0,42
20003 DATA 0,0,0,0,0,0,0,0,0
20010 DATA 0,0,0,0,0,0,0,0,0,
   0,0,40,0,10,95,192,37,93,
   192,38,235
20011 DATA 112,37,159,176,
   157,103,236,149,86,236,
   159,94,220,158
20012 DATA 93,152,158,85,158,
   37,182,222,37,93,185,39,
   85,117,39,231,93
20013 DATA 9,219,84,9,85,88,
   2,165,96,0,10,128,0
20020 DATA 0,112,0,1,248,0,3,
   240,0,7,228,0,15,204,0,15,
   224
20021 DATA 0,31,254,0,28,255,
   128,28,255,224,28,251,240,
   12,248,255,12
20022 DATA 248,63,7,240,62,7,
   224,60,3,192,56,1,128,48,
   1,128,0
20023 DATA 1,128,0,1,128,96,
   1,255,224,1,255,192,0
20030 DATA 0,112,48,1,248,56,
   3,240,60,7,228,62,15,204,
   127,15,225
20031 DATA 255,31,255,240,28,
   255,192,60,255,0,110,64,0,
   207,32,0,15
20032 DATA 152,0,7,240,0,7,
   224,0,15,192,48,24,96,48,
   48,48,112
20033 DATA 96,24,224,96,13,
   192,62,7,128,31,131,0,0
20040 DATA 1,57,0,1,57,0,2,
   56,128,2,124,128,2,56,128,
   1,57,0,1,17,0,0,40,0,0,16,
   0
20041 DATA 0,0,0,0,0,0,0,0,0,
   0,0,0,0,0,0,0,0,0,0,0,0,0,
   0,0,0,0,0,0,0,0,0,0,0,0,0
20042 DATA 0,0
20043 DATA 13,195,112,13,195,
   112,13,195,112,13,195,112,
   13,195,112,13,125,112,53
20044 DATA 85,92,53,105,92,
   53,170,92,53,170,92,213,
   170,87,245,85,95,229,85,91
20045 DATA 239,125,251,251,
   125,239,59,255,236,15,255,
   240,12,60,48,12,0,48,55
20046 DATA 0,220,213,195,87,
   0,0,0,0,0,8,0,2,8,8,0,8,
   32,0,42,128,0
20047 DATA 174,160,2,239,128,
   42,255,160,11,255,224,2,
   255,224,2,255,128,2,190
20048 DATA 0,8,186,128,0,184,
   32,0,40,0,0,32,0,32,32,0,
   0,0,128,2,0,2,0,0,0,0,0,0,
   0
20050 :
20051 REM ######## UDG DATA
   ########
20052 :
20060 DATA 3,3,15,31,63,126,
   124,124,192,192,224,240,
   248,126,62,62
20061 DATA 255,63,15,15,7,7,
   3,3,255,251,240,240,224,
   224,192,192
20062 DATA 128,192,192,227,
   244,248,248,248,1,3,3,199,
   47,31,31,31
20063 DATA 204,204,204,204,
   204,204,204,204,255,255,
   85,85,34,0,0,0
20064 DATA 16,59,255,255,255,
   255,255,255,255,255,255,
   255,255,255,255,255
20065 DATA 153,90,36,24,24,
   36,66,129,24,24,36,90,153,
   36,36,36
```

OOPS!

Your Commodore comes clean on its errors.

HERE AT YOUR COMMODORE WE pride ourselves in the quality of the listings that we print. Obviously we try to make sure that all programs are correct but occasionally errors do slip through, usually because they occur at stages of production that are out of our control.

This is the page where we come clean and give you details of errors that have appeared in recent issues.

MACH

The author of this program made some late changes to the actual listings. This meant that the end addresses for each part should have been altered. Unfortunately we did not spot this. Below you will find the start and end address for each part. Don't forget you can save the program from within the monitor with the S command.

Monitor – $8200 to $8E75
Macro – $9000 to $9C30
Assembler – $9000 to $9DB8

Some people are also experiencing problems with the .SER command in the Macro program. It appears that any labels generated do not increase in numeric order, but rather graphic characters are placed in the label. This only happens on some machines and does not effect the operation of the program as each label is still unique. However the author of the program is looking into this to see if he can find the problem.

March 1986

A couple of weird characters appeared in the listing of Kung Fu Masters. The characters should be the Commodore key and the @ sign when you see ñ, and a space when you see the á symbol. Also a number of lines were missing from the instructions. The missing lines are printed below.

PROGRAM: K.FU.INSTRUCTIONS

```
11850 PRINT"[s T]HE [s E]VIL
 [s R]EVOLUTION.  [s G]ROUP
S  OF  [s E]VIL"
11860 PRINT"FIGHTERS JOINED
 TOGETHER  TO  OVERCOME"
11870 PRINT"ANYTHING WHICH S
TANDS IN  THEIR  PATHS."
11890 PRINT"[s E]VEN THE [s
E]MPIRE'S TROOPS ARE FAILING
 TO"
11900 PRINT"RETURN AFTER THE
IR QUEST TO RID  THESES"
11910 PRINT"EVIL FIGHTERS FR
OM  THE  COUNTRY.  [s M]ANY"
11920 PRINT"HAVE TRIED BUT F
EW EVER RETURN TO  TELL"
11930 PRINT"OF THE EVIL GROU
PS HIDING PLACES."
11940 PRINT"[s H]OWEVER, FOR
 THE DEATH OF A FIGHTER,YOU"
```

A couple of wierd symbols also appeared in the Grid search listing. Whenever ' appears, replace it with a space and when \ appears replace it with a £ sign.

April

Because we didn't have a working Plus/4 we had to ask the author of the C-16/Plus/4 character generator to provide the listing. Unfortunately his listing was incorrect. Below you will find the lines that need to be changed.

PROGRAM: C16 CHAR.GEN

```
1070 DEF FNC(Z)=FNS(Z)-1024
1080 CH=0:TED=65280:SC=1:SL=
4:BC=9:BL=4:CC=1:CL=7

1150 ZZ$="":FORI=1TO6:READA:
ZZ$=ZZ$+CHR$(A):KEYI,CHR$(A)
1280 DO:GETA$:LOOP WHILE A$=
1300 LOOP UNTIL INSTR(ZZ$,A$

1920 PRINTSPC(20)"D MULTI OF

2040 FORR=0TO7:A=PEEK(14848+
R+(CH)*8)

2150 DO:GETA$:LOOP UNTIL A$=
"Y" OR A$="N"

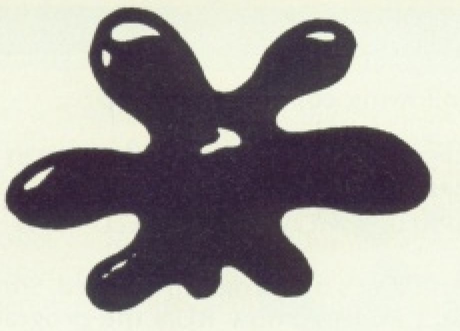
2210 DO:GETA$:LOOP UNTIL A$=
"Y" OR A$="N"
2280 DO:GET A$:LOOP UNTILA$=
"Y"ORA$="N"

2380 DO:GET A$:LOOP UNTILA$=

2530 DO:GET A$:LOOP UNTIL A$
="T" OR A$="D"
2590 FORI=1TOLEN(FM$):POKE51
1+I,ASC(MID$(FM$,I,1)):NEXT

2840 PRINT"[WHITE]"RIGHT$("
 "+STR$(I),2);"[BLACK]=
```

Some people are also experiencing problems with TOPMON in the same issue. This is because you have to move Basic before you RUN the program. The POKES given to perform this move are correct but if for some reason the location before the new start of Basic is not set to a zero you will get a syntax



error. Modify the given POKE command to:

POKE43,0:poke44,32:POKE8191,0:NEW

and you should have no problems.

The author of the article '2 for the C128' made a mistake with the positioning of the arrow in PROGRAM 2 line 220. The arrow should point to the 66 not the 42.

May 1986

Unfortunately the gremlins crept into the layout of the Programming the C-16 article. The last section of the program SYNTHDATA was placed at the end of the COMPILER by mistake. The COMPILER should only go up to line 45150 and lines 20080 onwards should follow line 20070 of the SYNTHDATA listing. Also line 55020 of the DEMO TUNES has a ',2' missing from the end.

Lines 50500 to 50530 are badly printed in some issues. Here are the lines that you may be having problems with.

PROGRAM: C16 DEMO TUNES

```
50500 DATALA,JOESTHEME,TW,0,
VD,7,DE,2,GO,LA,JOE,DE,0,GO,
LA,JOE,PA,100,GO,LA,HOVIS
50510 DATALA,JOE,C2,345,30,6
85,10,704,20,739,C2,169,20,5
96
50520 DATAC2,383,30,704,10,7
39,20,770,C2,262,20,643
50530 DATAC2,383,30,704,10,7
39,20,770,C2,262,20,643
50540 DATAC2,453,30,739,10,7
70,40,798
```

In the same issue a 1 was missed from the end of line 2780 of the COMMS GEN 5 program in the Telcom 64 article. This line should end with 371 NOT 37.

POLAR PETE suffers from a couple of problems. Firstly a couple of digits have been swapped in the POKE statement that you must enter before you RUN the programs. This should start with:

POKE16384 NOT POKE 16483.

Also a line was badly printed in Pete Load 3. The line should be:

2040 DATA141,3,212,169,32,141,2,212,96,169,13,141,5,212,169,10,1727.

WORDPROK suffered from duplication. On page 71 the listing is duplicated from page 70. You can ignore the listing on page 71 and the very last line on page 70 that starts <> -128.

As far as we are aware these are all of the problems in recent issues. Obviously we are trying to get listings more accurate each month. If in the meantime you do have problems then please do write in and we will deal with your query as soon as possible.



Stuart Cooke takes a look at a package that makes games design easier.

AFTER PLAYING YOUR THREE THOUSANDTH game of 'Blast the Alien' you'll probably think that you can do more with your C64 than just play games. Who knows you may even get the urge to write your own games to occupy yourself? However, there is one very big problem – programming. If you've ever tried to move an object around your C64's screen you will have found that it's not quite as easy as the professional programmers make it look. What you need is some sort of tool that takes the hard work out of games design. Well Activision has come to your rescue with Game Maker.

Game Maker is actually a number of tools, each one is aimed at helping you to design and write your games program. The Scene Maker lets you draw the background for your game. Sprite Maker allows you to design the characters that will appear on your screen. Music Maker and Sound Maker allow you to add music and sound effects to your program. The Editor allows you to 'write' a program that links all the above parts together to form a game.

Write is probably not quite the correct word for the way in which you write programs. You very rarely need to actually type anything on the keyboard as all of the programming commands are selected from the screen with a joystick. If the commands need any parameters then these are also selected with a joystick. For example if sprite number one was a dog then you would select the 'Sprite 1 is' command. The computer would then get the catalog of available sprites from your disk drive, selecting the dog sprite is simply a matter of moving the joystick until the word 'DOG' appears in the window. You have now told the computer that 'Sprite 1 is DOG'. Simple isn't it?



Some people may say that if you are going to have to program the game anyway why bother using a program such as this? Why not just write your program in Basic or machine code instead? If we take a look at the example program that you build up over the first few pages of the manual the reason for using a program like this should become apparent.

Let's assume that we have used the different 'Maker' programs to create a dog sprite, a jungle background and a piece of music, in this case the William Tell overture (Wiltel). Now lets have a look at a simple program. Firstly we want to display the background on the screen. The instruction:

SCENE 1 IS JUNGLE 2

I would do this. JUNGLE2 is the name used to save the scene on to disk. Now let's add the DOG sprite at position 40,154 these are the X and Y co-ordinates on the screen. Firstly we need to select the sprite and then position it. This is done with:

SPRITE 1 IS DOG
SPRITE 1 X POSITION = 040
SPRITE 1 Y POSITION = 154

Simple isn't it? Now let's animate the dog and move it across the screen. The following commands do this:

SPRITE 1 ANIMATION SPEED = 030
SPRITE 1 DIR = 064 RIGHT
SPRITE 1 MOVEMENT SPEED = 045

And that's our program finished. Simple isn't it? If we now RUN the program a dog will run across our jungle scene from left to right.

The reason for using a program like this should now be apparent, it is so simple to use.

Obviously there are bound to be limitations with a program like this. The author of Game Maker can hardly be expected to cater for every little effect that every programmer will need. Some of the major limitations are that firstly you can only have two screens to your game and secondly, you can only have eight sprites on the screen at once. If you were writing the program in Basic or machine code you could have as many screens as the computer's memory will allow. You can also get more than eight sprites on the screen at once through careful programming.

Even so it is possible to write a wide variety of 'games' with this program. Games provided on the disk with the program range from 'Chopper, a shoot'em up game to an animated Christmas card, complete with clockwork soldier and Jack in the Box.



The Editors

Obviously the quality of the games that you design with this program depends very much on how good the various Editors are. Well you will be pleased to know that they are all excellent. In fact some of them are better than some stand alone programs that are available. Because the programs are so powerful it is probably worth dealing with each one in turn.

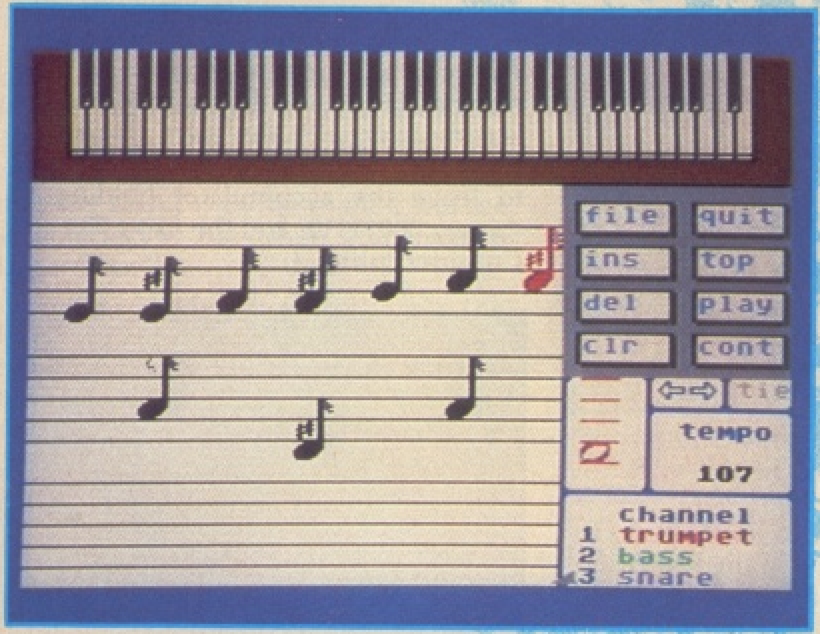
Scene Maker

This is the program that allows you to design your back drops. If you have ever

used a graphics program then you will recognise most of the available commands. Draw allows you to sketch on the screen in one of the available colours. You can have four different colours on the screen at any one time. All drawing is carried out via the joystick. The functions of Line, Box and Circle are quite obvious and Fill allows you to colour in areas of the screen. It is possible to copy areas of the screen from one position to another as well as Zoom in on a specific area so that it is easier to add fine detail.

One interesting thing about this program is the way that the menu of commands covers the top half of the





screen. The bottom half of the screen is used for drawing. My first thought on trying to draw my own background was how on earth do you draw on the top half of the screen? Then I realised that the Move command scrolls the bottom window so that you can see any area of the background scene. If you wish to see the whole picture, the View command turns off the menu.

Scene Maker is extremely well designed and very simple to use.

Sprite Maker

As you are no doubt aware, probably the most important part of any game is the sprites. These are the objects that move around the screen, shooting at you, killing you and generally getting in your way. Obviously a good game therefore depends on good quality sprites which are clear and well animated. The Sprite Maker program is extremely powerful, in fact I would go as far to say that it is one of the best sprite editors I have come across and it would stand up well on its own.

Sprite maker has provision for either multi-coloured or single colour sprites. Horizontal and vertical magnification is provided and up to four sprites can be placed edge to edge to create one large object that can be easily manipulated. Once you have defined your basic sprite you can then go on to alter it slightly and store these new 'pirctures' on a different frame. You can then run through the frames, like a piece of film, and make your masterpiece animate.

All of the details about the size of the sprite and the number of animation frames that it has are stored with the sprite when you save the sprite to disk.

Making Music

The music maker allows you to compose your own tunes for inclusion in your games. Music is entered under joystick control on to what is referred to as a musical sheet. This consists of three musical staves upon which you can place the note for one of the three available voices. Thirteen different instruments are provided ranging from snare to harpsichord and any voice can play any instrument.

Entering music is very simple. Select the voice that you wish to use. Select the length of the note and then place the note in its position on the music sheet.

If you have no musical knowledge at all then my feeling is that you may have some problems entering your own tunes. You could always try the 'suck it and see' approach and try moving notes around and listening to what results you get. Thankfully there are a fair range of tunes supplied on the Game Maker disk ranging from William Tell to Happy Birthday so there should always be something to suit your game.

Sound maker is the part of the package that lets you play around with a 'real' synthesiser. Again the program is under joystick control and everything is easy to alter. You can alter the type of wave form that you are using, alter the attack, decay, sustain and release of the volume and put the sound through filters. If you don't know what any of the above parameters are then I suggest that you just play around, twiddle the knobs and see what comes out. It is possible to link together a number of sound to create one effect. For example one sound may be a falling tone, another may be an explosion. Stick them together and you've suddenly got bombs falling from your aircraft.

And on it Goes

As I have previously said, this package does have its limitations and obviously you can't expect to get programs of really high quality out of it. However, it is a very good step up the ladder of designing your own games. Even if you can't program, the package will allow you to create those masterpieces that you so far have been unable to do anything about. Not only does this package let you try ideas out but it will also allow you to grasp what exactly goes into making up a game so that when you eventually go it on your own and try to write your own programs in either Basic or machine code you'll know exactly what you must do.

Game Maker is an extremely well thought out and easy to use package. The documentation is simple enough so that a fairly new computer owner could be writing games in a few house. All that is needed is a good imagination.

If you do have some ideas for games then this is a package that you should have in your collection.

WELCOME TO THE MACHINE

16 bit numbers are the subject of Allen Webb's foray intro machine code.

ARE YOU READY FOR THE DELIGHTS of 16 bit arithmetic. Even if you're not, that's what I intend to deal with in this article.

Up to now we've struggled within the constraints of eight bits, which, as you will have realised can become very messy.

If we work with 16 bit resolution, suddenly everything is simpler. You will recall that to increase a value by one, we can use the INC instruction. Consider Listing 1:

Listing 1

```
10 ASSEMBLE 90,I
90 REM *=$C000
100 REM INC 900
110 REM BNE LOOP
120 REM INC 901
130 REM .LOOP: RTS
140 REM ]
150 FOR I=OTO300
160 SYS 12*4096
170 PRINTPEEK (901) *256+PEEK (900)
180 NEXT
```

This increments a 16 bit number stored in locations 900 and 901. The low byte, 900, is first incremented in line 100. Line 110 checks to see if the low byte has reached zero. If it has, we need to increment the high byte. This will be obvious if you consider what happens as the low byte approaches zero:

Low byte	High byte	Combined value
254	0	254
255	0	255
0	1	256
1	1	257

The combined value is the low byte plus 256 times the high byte.

To reduce a memory location, you may recall that DEC is used. Listing 2 does this for a 16 bit number:

Listing 2

```
10 ASSEMBLE 90,I
90 REM *=$C000
100 REM LDA 900
110 REM BNE LOOP
120 REM DEC 901
130 REM .LOOP: DEC 900
140 REM RTS
150 REM ]
160 FOR I=OTO300
170 SYS 12*4096
180 PRINTPEEK(901)*256+PEEK(900)
190 NEXT
```

The principle of this routine is slightly different. The first step is to check whether the high byte needs reducing. This is done in line 110 which checks for a zero value in the low byte. Line 130 decrements the low byte every time.

You will have noted that there is an intimate relationship between the two bytes in a 16 bit numbers. Before we look at this, consider what happens when you add two numbers in decimal. If the two numbers add to above 10, you carry tens. Remember? Consider this sum:

```
19+
 3
22
```

The computer works in the same way in that if the result of adding two numbers exceeds 255, you have a carry to tell you about it. This effectively gives you nine bits and the carry flag is used to tweak the high bit. Addition with carry is performed by the instruction ADC.

Imagine that you wish to add the numbers 7 and 22. In binary these are:

```
   %00000111    7
+  %00010110   22
   %00011101   29
```

The result is less than 255 so we can still use an eight bit number. Consider the addition of $E4 and $3A:

```
   %11100100    $E4
+  %00111010    $3A
  %100011110   $11E
```

Suddenly, we have a number greater than 255 and the second byte must be used. The number is split as:

%00000001 and %00011110

In practical terms, how do we perform this addition? Consider this sequence of instructions:

```
CLS
LDA # $E4
ADC # $3A
```

The first step is to clear the carry flag — this prepares the way for the carry. The accumulator is then loaded with the number $E4. Lastly, $3A is added to the accumulator with the carry flag set as required. The result of this sequence is to leave the accumulator holding 30 (binary 00011110) and the carry flag set. Consider listing 3:

Listing 3

```
80 ASSEMBLE 90,I
90 REM *=$C000
100 REM LDA # 0
110 REM STA 902
120 REM STA 903
130 REM CLC
140 REM LDA 900
150 REM ADC 901
160 REM STA 902
170 REM LDA 902
180 REM ADC # 1
190 REM STA 903
200 REM .LOOP; RTS
210 REM ]
220 INPUT "A,B";A,B
230 POKE 900,A: POKE 901,B
240 SYS 12*4096
250 PRINTPEEK(902)+PEEK(903)*256
```

This adds two numbers in 900 and 901 and puts the result in the 16 bit number in 902 and 903. Lines 100 to 120 clear the 16 number and lines 130 to 150 add the two numbers. Line 160 stores the low byte of the result and lines 180 to 190 store the carry in the high byte.

Subtraction is a simlar process albeit slightly trickier to understand. The relevant instruction is SBC (Subtract with Carry) and again the carry flag is used to convey information on the result of the subtraction. This time, the carry is cleared if the subtraction results in an underflow (negative number). Don't worry about the mechanism of SBC this time, I'll deal with twos complimenting and other mysteries in the future. For now, simply accept that SBC is the reverse of ADC. Consider listing 4.

Listing 4

```
80 ASSEMBLE 90,I
90 REM *=$C000
100 REM SEC
110 REM LDA 901
120 REM SBC 900
130 REM STA 901
200 REM RTS
210 REM ]
220 INPUT "A,B";A,B
230 POKE 900,A: POKE 901,B
240 SYS 12*4096
250 PRINTPEEK(901)
```

The first step, Line 100, is to set the carry flag. The accumulator is then loaded with the first number and Line 130 subtracts the second number. The result is put into location 901. Try messing about with the routine and see what effect it has when you make B larger than A.

Let us consider a useful example. The first answer to your last month's homework was rather tacky answers at the end of this article. Listing 5 gives a method using 16 bit arithmetic.

Listing 5

```
80 ASSEMBLE 90,I
90 REM *=$C000
110 REM LDA#0
120 REM STA $FB
130 REM LDA#04
200 REM STA $FC
210 REM LDY#0
220 REM .LOOP: LDA #42
230 REM STA ($FB),Y
250 REM INC $FB
260 REM BNE LOOP1
270 REM INC $FC
280 REM .LOOP1: LDA $FB
290 REM CMP # $E8
300 REM BNE LOOP
310 REM LDA $FC
320 REM CMP#7
330 REM BNE LOOP
340 REM RTS
350 REM ]
```

The key to the routine is the 16 bit number in locations $FB and $FC. Instead of varying the Y register 10 index this address, we will keep it set to zero and alter the base address. Lines 110 to 200 set the address to the start of the screen ($0400). We then zero the Y register. The main loop puts an asterisk at the currently addressed location (Lines 220 to 230). Lines 250 and 270 increment the base address by one. Lines 280 to 330 compare the base address to 807E8 (the last address of the screen) and loop back if it hasn't been reached. Since we're dealing with 16 bits, two comparisons (Lines 290 and 320 are required). This is clearly a more satisfactory way of working.

Finally, Listings 6 and 7 give routines for the addition and subtraction of two 16 bit numbers. One number is in locations 900/901 and the other in 902/903. The resulting number is left in locations 900/901.

Listing 6

```
80 ASSEMBLE 90,I
90 REM *=$C000
110 REM CLC
120 REM LDA 900
130 REM ADC 902
200 REM STA 900
210 REM LDA 901
220 REM ADC 903
230 REM STA 901
240 REM RTS
350 REM ]
360 INPUT "A,B"; A,B
370 POKE901,A/256:POKE900,A-PEEK(901)*256
380 POKE903,B/256:POKE902,B-PEEK(903)*256
390 SYS 12*4096
400 PRINTPEEK(901)*256+PEEK(900)
```

Listing 7

```
80 ASSEMBLE 90,I
90 REM *=$C000
110 REM SEC
120 REM LDA 900
130 REM SBC 902
200 REM STA 900
210 REM LDA 901
220 REM SBC 903
230 REM STA 901
240 REM RTS
350 REM ]
360 INPUT "A,B";A,B
370 POKE 901,A/256:POKE 900,A-PEEK(901)*256
380 POKE 903,B/256:POKE 902,B-PEEK(903)*256
390 SYS 12*4096
400 PRINTPEEK(901)*256+PEEK(900)
```

I now want to briefly discuss an alternative way of manipulating numbers. Consider the binary number seven:

%00000111

If the bits are shifted left one place with the left-most bit lost and the right-most bit set to zero, we get:

%00001110

or the number 14. What we have done is multiply the number by two. Similarly, if you shift the bits right one place, we divide by two. The instructions ASL (Arithmetic Shift Left) and LSR (Logical Shift Right) perform these functions. To multiply a number in location 900 by two, you simply use:

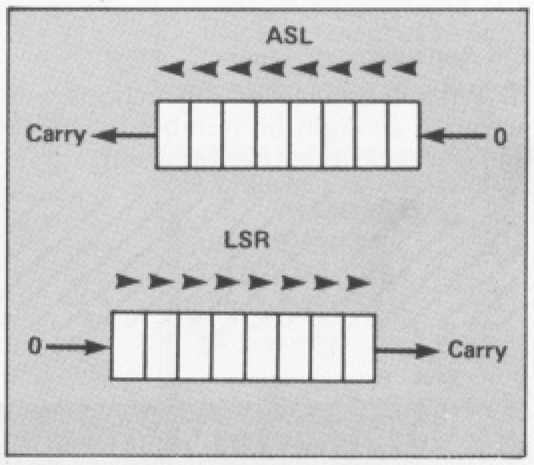
ASL 900

to multiply by four, use:

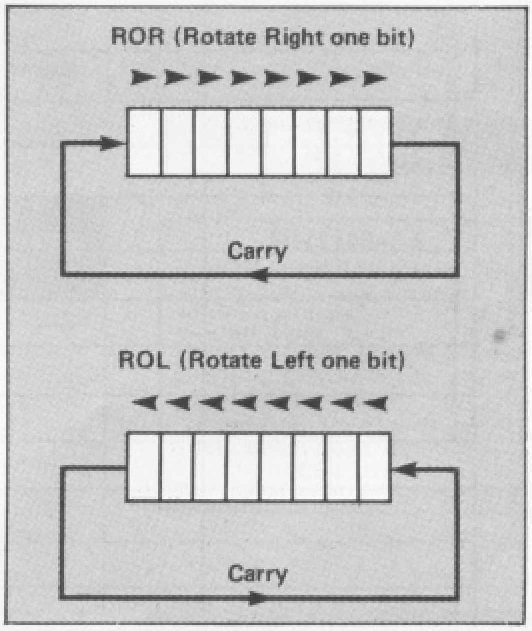
ASL 900
ASL 900

and so on.

In fact, these instructions do not lose the end bit. As they shift the bit is pushed into the carry flag:



To allow you to make use of the carry flag to manipulate 16 bit or larger numbers, there are a further two instructions:



These rotate the bit pattern but incorporate the carry bit into the number. Consider the pair of bytes:

BYTE 1	**BYTE 2**
00000000	10101010 = 170

Let us shift byte two left once and then roll byte one left once. The left-most bit

of byte two moves into the carry flag and is then moved into byte one giving:

BYTE 1	BYTE 2
00000001	01010100 = 340

Voila! We've multiplied a 16 bit number by two. The general methods of doing this are:
Multiply by 2:
ASL Low byte
ROL High byte
Divide by 2:
LSR High byte
ROR Low byte
The shift and roll instructions operate on the accumulator and in absolute, zero page, absolute,X and zero page,X addressing modes. We'll meet all of these commands again so dont fear if you haven't fully grasped them.

OK, pain time:

1) What sequence of instructions will multiply an eight bit number by 256 (a bit of a trick question this one)?

2) Write a simple routine to increment a 16 bit number of 40.

3) Starting with a 16 bit number in 900/901 (number A) and a 16 bit number in 902/903 (number B), write a routine which will perform the calculation 2*A+B/2 and store the result in locations 904/905.

Next time, more arithmetic mysteries, the BIT instruction and a bit of Boolean.

```
10 ASSEMBLE 90,I
90 REM *=$C000
100 REM LDY # 0
110 REM LDA # 42
120 REM .LOOP: STA 1024,Y
130 REM STA 1274,Y
140 REM STA 1524,Y
150 REM STA 1774,Y
160 REM INY
170 REM CPY # 250
180 REM BNE LOOP
190 REM RTS
200 REM ]
```

The second routine copies the top line of the screen to the 20th line.

```
10 ASSEMBLE 90,I
90 REM *=$C000
100 REM LDY # 0
120 REM .LOOP: LDA 1024,Y
130 REM STA 1824,Y
140 REM INY
150 REM CPY #40
160 REM BNE LOOP
170 REM RTS
180 REM ]
```

Foot Note

Due to a problem with the printer used for this series, the # characters in previous articles have been replaced by a £ symbol.

We hope this won't happen in the future and that it hasn't caused too many headaches.

YOUR COMMODORE

Lineage: 40p per word. (46p incl. VAT).
Semi display: £9.00 per single column centimetre (plus VAT). Ring for information on series bookings/discounts.

All advertisements in this section must be prepaid. Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).

01-437 0699
EXT. 291

Send your requirements to:

ASP LTD, 1 GOLDEN SQUARE, LONDON W1

SOFTWARE

C16/PLUS 4 SOFTWARE

Title	RRP	OUR
Commando	7.95	**6.75**
Games Designer	9.95	**8.50**
A.C.E. (C-16)	9.95	**8.50**
A.C.E. (Plus 4)	10.95	**9.40**
Airwolf	6.95	**5.95**
Steve Davis Snooker	7.95	**6.75**
Daley Thompson Star Events	6.95	**5.95**
Jump Jet	9.95	**8.50**
Games Pack I (15 games)	5.95	**4.95**
Games Pack II (15 games)	5.95	**4.95**
World Cup Football	6.95	**5.95**
Gullwing Falcon	6.95	**5.95**
The Boss	6.95	**5.95**
Anirog Favourite Four	7.95	**6.75**
C-16 Classics	9.95	**8.50**
C-16 Classics II	9.95	**8.50**
Gremlins	7.95	**6.75**
Spiderman/Hulk Pack	7.95	**6.75**
Atlantis	6.95	**5.95**
Grand Master Chess (C-16)	8.95	**7.75**
Grandmaster Chess (Plus 4)	8.95	**7.75**
Tom Thumb	6.95	**5.95**
Sword of Destiny	6.95	**5.95**
Xargons Revenge	6.95	**5.95**
World Series Baseball	7.95	**6.75**
Robcon Head Alignment	4.94	**4.50**
Beach Head	7.95	**6.75**
Defence 16	7.95	**6.75**
Dark Tower	5.95	**4.95**
Wizard and Princess	5.95	**4.95**
Classic Adventure	5.95	**4.95**
Supergran	9.95	**8.50**
Canoe Slalom	6.95	**5.95**

BUDGET SOFTWARE
Mastertronic Titles (£1.99 each)
Formula 1 Simulator
Vegas Jackpot
Tutti Fruity
BMX Racers
Rockman
Squirm
Spectipede
Big Mac
Mr Puniverse
Buy any 3 for **£5.25**

Solar Titles (£2.99 each)
Knock-Out
Mission Mars
Suicide Run
Alien Attack
Cruncher
Quick Draw
Buy any 3 for **£7.95** or Buy any 2 for **£5.45**

C-16 JOYSTICK £9.95
Please state which micro.

All prices include postage and packing in the U.K.
Payment with order by cheque or postal order please to:
JOTHAN SOFTWARE (Dept Y.C.1)
65 Heol Gwys, Upper Cwmtwrch, Swansea SA9 2XQ. Tel: 0639-830934

IMPACT, for the Commodore 64/128 is a new monthly computerised magazine, on cassette only. There's Reviews, Tips, Quizzes, etc. Plus a Free Text Adventure. All this for only £2.00. If you want to send in your own Letter, Tips, Review, etc, then just send £1.90 when you order impact. Impact 1 will be out in one month. Send Cheque/P.O., made payable to: M. Sibthorpe, 18 The Stray, Darlington, Co. Durham DL1 1EP.

---

64 SOURCE GENERATOR; an advanced utility that will produce source code from programs in memory: imagine being able to take any machine language program and convert it to true source code, complete with labels. SAE more info; £6.95 Tape: £8.95 Disk. Cheques/POs to K. Otton, 19 Meadows Close, London E10 7DH.

DERBY DAY
C16/14 **£5.75** **CBM 64** **£6.75**
*Visual Form Guide
*On Screen Action
*Realistic Betting
*Over 50 Horses
*10 Race Tracks
*4 Racing Distances
Bloomfield Software
PO Box 69, Blackpool FY2 9EX

SOUTH EAST KENT SOFTWARE
Dept Y/C, PO Box 152, Orpington, Kent.
We stock for Commodore.
Please state which model.
Send S.A.E. for full price list.

COMMODORE 64. GAMES
Silent Service, Biggles, Superbowl, Ghosts 'n' Goblins, Gladiator, Mugsys Revenge, Way of the Tiger, Bomb Jack, Commando, Back to the Future, Winter Games, Summer Games II

BUSINESS & EDUCATION
Superbase 64, Superscript, Home Accounts, Maillists, Stock Control, VAT File

HARDWARE
Joysticks, Music Makers.

SOCCER MANAGER — Manage your own team, 4 divisions, F.A. Cup, transfers tactics etc.
RACENIGHT — Run your own racenight, up to 10 players. Exciting horse racing game.
SOCCER QUIZ — Test your soccer knowledge, hundreds of questions.
Any one for **£3.99**, 2 for **£6.99**, 3 for **£8.50**
T. PALMER, 59, Old Park Road, Sheffield 8. Tel: 0742 749498.

ALPHASOFT
SPECIAL OFFERS
Hardball **£6.99** Fairlight **£6.99**
Psi-5 Trading Co. **£6.99** Superbowl **£6.99**
Law of the West **£6.99** V **£6.50**
Zoids **£6.50** Bomb Jack **£6.99**
Z Zap Sizzlers (4 Top Titles) **£6.99**

AND MANY MORE!
Cheque/P.O. to **ALPHASOFT**
115 Parsonage Road, Withington, Manchester M20 9WZ
(S.A.E. for Price List. P&P included)

Genealogy Program
for the 64 & 128 **"FAMILY TREE"** will produce Pedigree Charts, Family Group Records, Individual Files, Indexes, Searches of Ancestors. LDS version available.
£32.95 including air mail service.
GENEALOGY SOFTWARE
1046 Parkwood Ave, Sarnia, Ontario, Canada N7V 3T9

Animal Records
maintained with **"PEDIGREE"** for the 64 & 128. Produces Litter, Awards, Breeding, Show, and individual records, and Pedigree Charts. **£44.95** including air mail service.
GENEALOGY SOFTWARE
1046 Parkwood Ave, Sarnia, Ontario, Canada N7V 3T9

ADVERTISE IN YOUR COMMODORE PHONE 01-437 0626 FOR DETAILS!

REPAIRS

COMMODORE REPAIRS
Commodore repairs by Commodore approved engineers. Repair prices: Vic 20 modulator **£8.05**, Vic 20 **£15.00**, CBM 64 from **£20.00**, C2N from **£9.00**, 1541 from **£20.00**, C64 psu **£13.50**, Spectrum **£16.50**, Dragon **£32.00**, Electrons, Atari.
For more information SAE or telephone:
G.C. BUNCE & SON,
36 Burlington Road, Burnham, Bucks SL1 7BQ.
Tel: (06286) 61696.
9am-6pm Mon-Fri.

TRIDENT ENTERPRISES LTD
Commodore 64 — **£15.** Vic 20 — **£12.50**, Spectrum — **£10.50**, QL — **£30**, 1541 Disk Drives — **£20**, and all Commodore systems. Send faulty machine with note describing fault to:

UNIT 7, WENTWORTH INDUSTRIAL COURT, GOODWIN ROAD, BRITWELL, SLOUGH TEL: 0753 21391

Overseas enquiries welcome. (Above prices are exclusive of parts and VAT).

When replying to classified advertisements readers are recommended to take steps to protect their interest before sending money.

REPAIRS

LONDON COMPUTER REPAIR CENTRE
★ Spectrum, Commodore, BBC, Atari, Amstrad etc.
★ Low cost repairs from **£12.00**
★ Collection & delivery optional
★ Used Micro's bought, sold, exchanged.
Call **NIGEL: 01-863 7166**
MICRO SUPPORT
Unit 3, 15 Springfield Road, Harrow, Middx HA4 1QF.

The 1st choice
EXPRESS COMMODORE REPAIRS!
WHY PAY HIGH 'FIXED' PRICES
Commodore repairs between £14 and £35

FOR PROFESSIONALISM – "Of all the repair companies 'Sinclair User' spoke to, **MANCOMP** seemed the MOST PROFESSIONAL when it came to advice and a helpful attitude" AUG 85.

WHY DELAY – OUR ESTIMATES ARE FREE AND WITHOUT OBLIGATION. IF YOU DO NOT LIKE OUR ESTIMATE WE WILL RETURN THE COMPUTER **AND** REFUND THE POSTAGE INCURRED!

MANCOMP LTD (Dept YC5)
Printworks Lane, Levenshulme, Manchester M19 3JP
OPEN MON - SAT 9am to 7pm
061 224 1888/9886

EPROMS

PROM-64
EPROM programmer for 8K, 16K, ZIF socket, Vpp generator & software on tape **£34.75**
64-CARTridge, 2 × 8K **£5.95**
EPROMS QTY 1-10, 2716-128 **£2.99**
DHOBI-2 EPROM eraser **£22.95**
UK + 15% VAT
Cambridge Microelectronics
One Milton Road, Cambridge CB4 1UY
Tel: (0223) 314814 Tlx: 81574

SPORT

HORSE FORM
CBM 64/128. The ultimate in Horseracing Predictions. Why pay £5 for one-off time ratings when you can continually produce your own. Program holds data for all flat racing tracks in Britain and Ireland.
Get The Edge, Get Horseform.
£7.95 cassette. Cheques/P.O. to:
SEESAW SOFTWARE,
74 Fleetham Gardens, Lower Earley, Reading RG6 4BY.

WARNING NOTICE

Advertisements placed in this magazine are to be in strict compliance with our standard conditions (copies of which conditions are available on request) and on the clear understanding that the advertiser warrants that his advertisement(s) does not infringe any copyright or condition of sale of any interested party in the advertised product.

Further, the advertiser indemnifies the proprietors of this magazine in respect of costs, damages, or any other claims brought against them as a result of legal action arising from the publication of the advertisement.

Any breach of these terms or the said conditions may result in prosecution of the advertiser by the proprietors.

FOR SALE

GADGET 64 £6.95
Programming tool kit for CBM64 specially designed for tape users.
Provides: RENUMBER (inc. GOTO, GOSUB), APPEND, DELETE, and a special PARK facility (allows two programs in memory for editing, appending, etc.)
Send your name, address & PO/cheque for £6.95 (P&P free) to:

J&J GILMOUR LTD
Dept. YC586, 4 Copse Close, Liss, Hampshire GU33 7EW, England.
Tel: 0730-893430

DISCS

!!! FREE DISKS !!!
For every 10 ordered we supply 11

5¼"
SS/DD **£10.99** SS/QD **£13.19**
DS/DD **£13.19** DS/QD **£15.39**

3½"
S/Side **£27.49** D/Side **£32.99**

All prices **INCLUDE VAT**
Delivery is **FREE**
All disks guaranteed for 12 months
Cash with order please to
HOLLBARN Ltd., Dept CM1, 8 Alexander Road, Stotfold, Hitchin, Herts. SG5 4NA.

GENERAL

BUY OR SELL your unwanted software, books, and hardware. We put you directly in contact with each other in your area. Details: Second Hand Computer Sales Agency. 38, Bowling Alley Walks, Dorchester, Dorset. Tel (0305) 64849 (24 hrs).

WORD DEALER, an exciting word game for up to four players. Four play levels too! Send cheque/P.O. £3.25 to R. Porter, Woodside House, Abbey Road, Scone, Perthshire, PH2 6LL (C16 and plus/4).

HARDWARE

★ BACK-UP-SYSTEMS FOR CBM-64 & VIC-20 ★

The BACK-UP-BOARD Mk. 1 is the fastest, and easiest way to back up your tape programs. With this board you can connect a second CBM cassette deck to your computer. You can copy a program on the second deck while the original loads from the first. You can back up all fast loaders, data files, etc. BACK-UP-BOARD Mk. 1 £10.00 inc. P&P.

BACK-UP-BOARD Mk. 2. This new version of our famous BACK-UP-BOARD is now proving so popular it is outselling our original Mk. 1. This new board now works two ways. Switched to 'NORMAL' it operates as the Mk. 1. Switched to 'GENERATE', '64' owners can use our special transfer software to regenerate a copy of a program on the second deck as it loads from the first. This method can produce a better quality copy than the original. BACK-UP-BOARD Mk. 2 & Software £14.50 inc. P&P. Conversion service. Return your Mk. 1 Board, if purchased from us, to be modified to Mk. 2 for £4.50 including software & P&P.

AZIMUTH HEAD ALIGNMENT TAPE by INTERCEPTOR for '64'. Use this test tape with your '64' to adjust the heads of your decks to make a perfectly matched pair. Full instructions and screwdriver supplied. £8.50 inc. P&P.

ROM SWITCH/RESET BOARD. Fit this board in the expansion port of your '64' and you can reset ALL programs. Resets programs that a standard reset switch can't. A socket for a 2764 8K eprom is also fitted. Full instructions supplied. £8.95 inc. P&P.

Please send SAE for more details of the above products. Send your cheque/PO to:

TURBOTRONIC
46 Ripon St., Parkinson Lane, Halifax, West Yorkshire HX1 3UG.
Tel: (0422) 52020
(Europe add 50p outside Europe add £1.50)

FOR HIRE

HIRE HIRE

STAFFS COMPUTER LIBRARY
CBM 64 OWNERS

Join the friendly Library. We give you what you want. If a title is not on our list then we will get it, if it's still available. (Games only)

- All the latest Titles
- Full 7 Days Hire
- Low Hire Cost. From 75p
- Up to 4 Games per week
- All games original
- Lots of American Titles
- Low Hard and Software prices
- Membership ONLY £5.00

Free m/c Monitor on joining

Send your cheques/postal orders payable to:
STAFF COMPUTERS
The Newlands, Greensome Lane, Stafford

VIC-20/CBM-64 SOFTWARE HIRE

Free Membership. Hundreds of Titles to choose from. Send S.A.E. for your Free Hire Kit:
VSH (YM), PO BOX 65, NOTTINGHAM NG1

COMMODORE 64 cassette library. Hire latest games and educational software, only one price £1.50 per cassette 7 days hire. Details S.A.E. please: C.C.L., 19 Darrell Way, Abingdon, OX14 1HQ.

FOR SALE

MOONRAKER £7.95
Space Flight Game for the CBM64
Features: Animated displays of flight over surface craters. Phasor Ray to shoot alien craft, score and flight indicators. Sophisticated game using advanced display techniques.
Send your name, address & PO/cheque for £7.95 (P&P free) to:

J&J GILMOUR LTD
Dept. YC586, 4 Copse Close, Liss, Hampshire GU33 7EW, England.
Tel: 0730-893430

HARDWARE

Super Breaker Back-Up Board

For the Commodore 64, CBM 128 and VIC-20. When a second CBM type cassette unit is connected to the board it will make security back-up copies of cassette based software, including Turbo/Fast Loaders, Basic and/or Machine Code and Data Files. No additional software is needed. Very User Friendly, the back-up copy is made on the second cassette while the original is loading on the first. Price at £8.95 Plus P&P.

RESET CARTRIDGE for CBM 64. This cartridge will reset all, yes all, memory resident programs. Including all programs where standard reset switches fail. Price £7.95 + P&P.

P&P 55p UK (Europe add £1.00, elsewhere add £2.00). Send cheque/postal order, or a stamp for details to:

C. Kent, 26 Northcape Walk, Corby, Northants. NN18 9DQ.
For enquiries telephone (0536) 742622

SOFTWARE APPLICATIONS

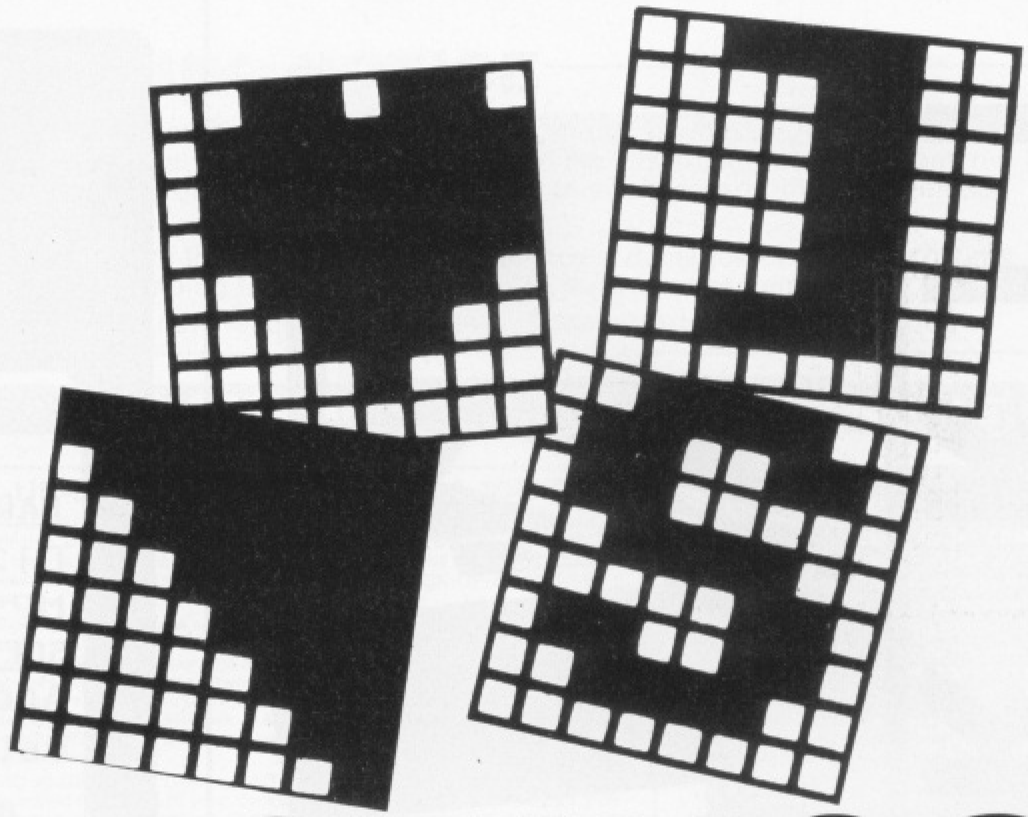
HOME ACCOUNTS

Put your house in order! Probably home computing's best use! Comprehensive coverage of bank accounts. Credit cards, HP in-built accuracy check. Records all transactions. Projects cashflow for any period ahead. Available for Complete Commodore Range. £8.45 or *free* details from:

Discus Software
Freepost
Beach Approach
Brixham TQ5 8BR
Tel: 08045 55532

ADVERTISERS PLEASE NOTE:—
COPY DEADLINES FOR FUTURE ISSUES OF YOUR COMMODORE ARE AS FOLLOWS:— SEPT ISSUE - 24th JUNE
OCT ISSUE - 28th JULY. NOV ISSUE - 26th AUGUST
WE WORK WELL IN ADVANCE SO START THINKING NOW!

Listings will be much easier to enter with our new system.



LISTINGS

COMMODORE LISTINGS ARE RATHER well known for the horrible little black blobs that always abound. Unfortunately the graphics characters which are used to represent graphic and control characters do not reproduce very well and they are also difficult to find on the Commodore keyboard.

In future all control and graphics commands will be replaced by a mnemonic within square brackets. This mnemonic is not typed out as printed in the magazine but rather the corresponding key or keys on the keyboard are pressed. For example [RIGHT] means press the cursor right key, you do not type in [RIGHT]. All of the keywords, what keys to press and how they are shown on the screen are shown below.

Any character that is accessed by pressing shift and a letter will be printed as [Sletter].

[SA] shift and A

[S+] shift and +

Any character that is accessed by pressing the Commodore key and a letter will be printed as [Cletter]

[CA] Commodore and A

[C+] Commodore and +

[C1] Commodore and 1

If any characters are repeated the mnemonic will be followed by a number. This number is how many times you should enter the character. Any number of spaces over one will also be represented in this form

[RIGHT10] press cursor right 10 times

[C+10] press Commodore and + 10 times

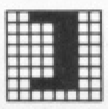
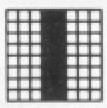
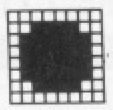
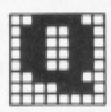
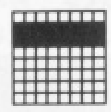
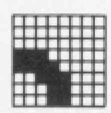
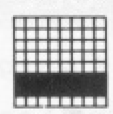
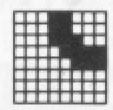
[SPC10] Press the space bar 10 times

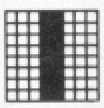
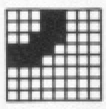
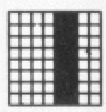
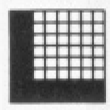
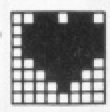
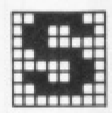
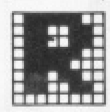
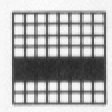
Any other characters should be easily recognisable for example CTRL-N means press CTRL and N and LEFT-ARROW means press the left arrow.

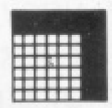
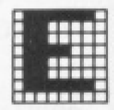
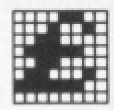
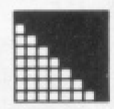
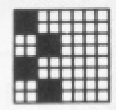
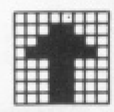
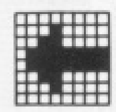
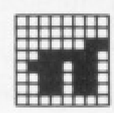
Any number of mnemonics can be enclosed in brackets for example

[SA10,SPC10,SA10]

means type 10 shift A's 10 spaces and another 10 shift A's.

Mnemonic	Symbol	what to press
[RIGHT]		left/right
[LEFT]		shift left/right
[UP]		Shift & up /down
[DOWN]		up/down
[F1]		f1
[F2]		shift & f1
[F3]		f3
[F4]		shift & f3

Mnemonic	Symbol	what to press
[F5]		f5
[F6]		shift & f5
[F7]		f7
[F8]		shift & f7
[CLEAR]		shift & CLR /HOME
[HOME]		CLR/HOME
[RVSON]		CTRL & 9
[RVSOFF]		CTRL & 0

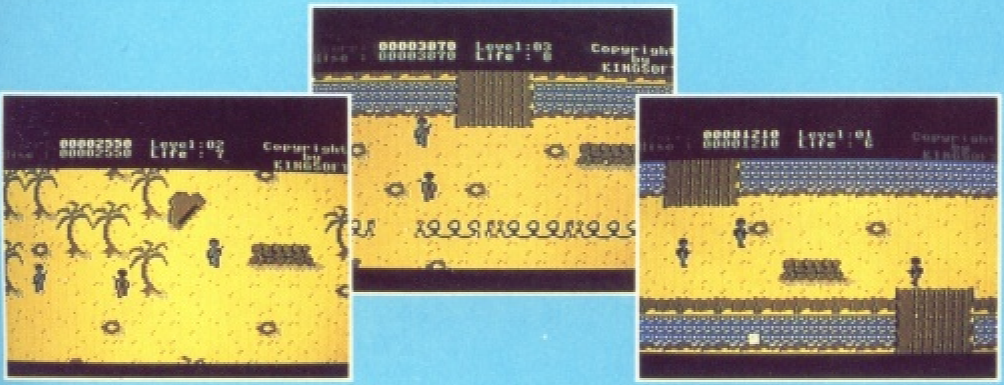
Mnemonic	Symbol	what to press
[BLACK]		CTRL & 1
[WHITE]		CTRL & 2
[RED]		CTRL & 3
[CYAN]		CTRL & 4
[PURPLE]		CTRL & 5
[GREEN]		CTRL & 6
[BLUE]		CTRL & 7
[YELLOW]		CTRL & 8

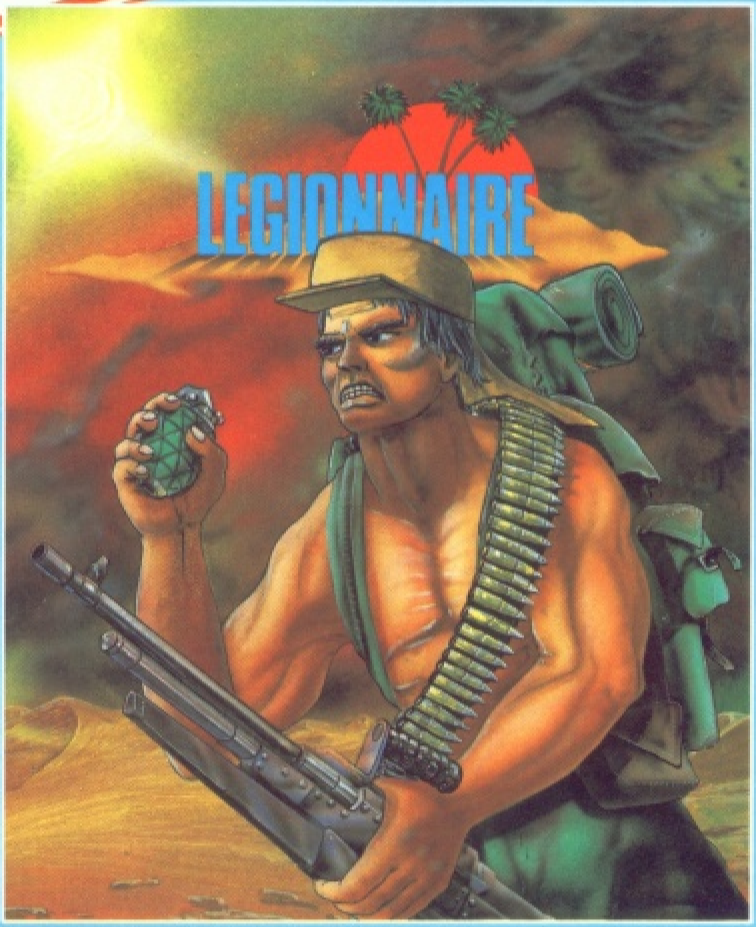
C16/PLUS 4

LEGIONNAIRE

Your mission is to take vital messages to the garrison headquarters. As you travel through the scrolling landscapes, you will be under constant fire from the tribesmen bent on reaching the fort. You will have to use your machine gun with great skill to get past them. There are seven separate areas of scrolling landscapes before the fort can be reached and each area requires increasing skill.

C16/PLUS 4 £5.95

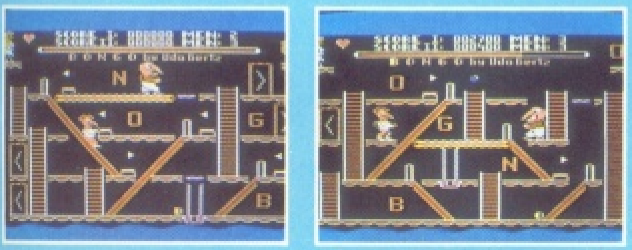




BONGO CONSTRUCTION SET

Bongo Construction Set is a hilarious family game. Bongo is trying to reach the princess, hotly pursued by monsters. He uses ladders, slides, transporters, lifts and trampolines to elude his pursuers but they seem to know his moves before he makes them. There are six stages with five skill levels. You can design your own stages using the joystick and save them on cassette or disk to play later.

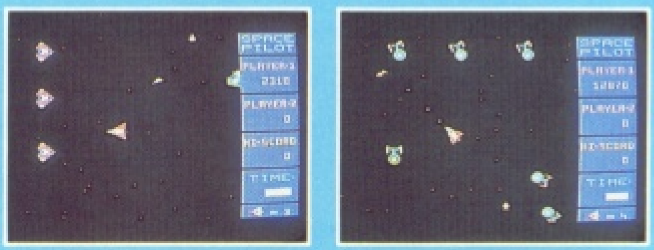
C16/PLUS 4 £5.95



SPACE PILOT

Space Pilot, the legendery hero, is under constant attack by alien ships with one aim - the destruction of his craft. The alien craft attack singly or in battle formation. Equipped with heat seeking missiles, their attack is deadly as well as relentless. Space Pilot with full 360 degrees scrolling and four stages, demanding ever increasing skills retains all aspects of a smash hit on CBM 64.

C16/PLUS 4 £5.95



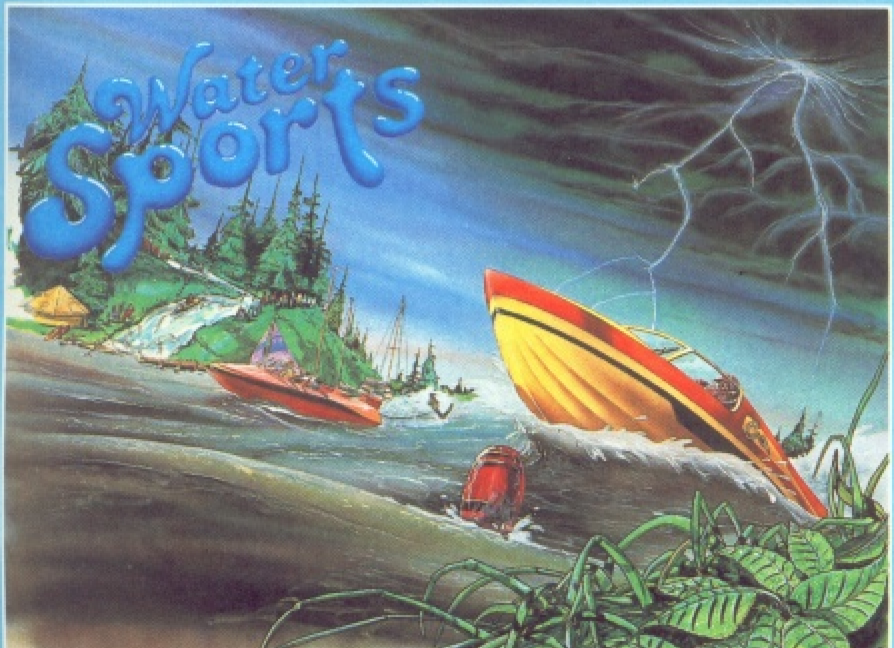
THAI BOXING

A unique combination of the noble art of Boxing and skills of Martial arts. Three stages of increasing skill.

16/PLUS 4 £5.95



FORTHCOMING ANCO RELEASES



WATER SPORTS
Water Skiing and Shooting the Rapids. C16/PLUS 4 £6.95

WATER SPORTS PLUS
Enhanced version for PLUS 4 owners or C16 with 16K expansion includes an extra event, Power Boat Racing. £7.95 PLUS 4/C16+16K.

CHAMPIONSHIP TENNIS
A Tennis Simulation quite capable of creating the drama of the Centre Court. Available June.
C16/PLUS 4 £5.95



MICRO TEXT
Ideal for home user. Facilities include Centring and full Left and Right justification.
£7.95 (Cass). £10.95 (Disc).

MICRO BASE
An easy to use data base with full sort facilities. Ideal for keeping track of records or stamp collection. Club secretaries will find it very handy for storing names and addresses of members. Designed to print self-adhesive labels. Selected names and addresses can be selected for printing.
£7.95 (cass). £10.95 (Disc).

C16 and PLUS 4 REFERENCE BOOK
The book has been specially written for ANCO' It is a comprehensive guide for a beginner and a professional programmer. June. £7.95

ANCO

ANCO SOFTWARE, 4 WEST GATE HOUSE, SPITAL STREET, DARTFORD, KENT. DA1 2EH.
Telephone: 0322 92513/92518.
Mail Order: Payment by Cheque, P.O., Visa or Access. 24 Hour Credit Card Service: 0322 522631.

Choosing the right computer is a good start — but can you find the right software?



At SUPERSOFT we're very conscious of the fact that people who spend several hundred pounds on computer equipment are looking to do rather more than play Space Invaders.

Financial planning is a rather grand name for something you've been doing all your life – making ends meet! Perhaps if Mr Micawber had used **BUSICALC** he would have been able to balance the books a little better.

For home, club or small business use BUSICALC 1 should pay for itself in no time at all; for larger companies we recommend BUSICALC 3, one of the few really valuable programs that you can learn to use in a day.

Although your Commodore 64 is a powerful musical instrument you need to be a pretty good programmer to understand how it all works. Unless, of course, you buy **MUSIC MASTER!**

To use MUSIC MASTER requires no prior musical knowledge, though in the hands of an experienced musician it will prove an invaluable tool. You don't need to know the first thing about programming either! MUSIC MASTER is the musical equivalent of a word processor, remembering the notes you play and allowing you to replay and edit them as you wish.

INTERDICTOR PILOT is a space flight simulator. Nowadays simulators are widely used to train pilots and astronauts because – to be frank – it's a lot cheaper (and safer) than the real thing!

Imagine, if you will, life in the 22nd century: space travel is commonplace, and on the outskirts of the galaxy the first war between civilizations is being fought. A shortage of trained pilots has prompted the Federation to develop a computer simulation that allows raw recruits to gain experience without paying for their mistakes with their lives. With the aid of your Commodore 64 you too can learn to pilot the Interdictor Mk 3 craft. But be warned – this is no game!

Other SUPERSOFT products include the **MIKRO ASSEMBLER** cartridge, the only assembler that's ideal for beginners yet powerful enough for the professional (most of our competitors use it!). The **VICTREE** cartridge adds dozens of commands to Basic including toolkit aids and disk commands; or on disk there's **MASTER 64,** a really comprehensive package for the keen programmer.

Of course, we do also publish games programs, and with classics like **STIX, QUINX** and **KAMI-KAZE** in our range we are one of the market leaders. But we most enjoy coming up with the sort of programs that are going to be in use for months and years, not hours and days – the sort of programs that make you glad that you bought a computer – and glad that you bought SUPERSOFT!

You won't find SUPERSOFT products on the shelves of your local supermarket. But most specialist shops stock titles from our extensive range (and are prepared to obtain other programs to order). However you can also buy direct by sending a cheque (pre-paid orders are post free!), by calling at our offices, or over the telephone using your ACCESS card.

SUPERSOFT

SUPERSOFT, Winchester House, Canning Road, Wealdstone, Harrow, Middlesex HA3 7SJ
Telephone: 01-861 1166